सत्यमेव जयते

वर्गीय अपेक्षाएँ के लिए मानक दस्तावेज़ सं:
टी.ई.सी. 91010:2026

TEC STANDARD FOR GENERIC
REQUIREMENTS No.
TEC 91010:2026

कवांटम सुरक्षित एवं किालसकि क्रिप्टोग्राक्रिक प्रणालियाँ

QUANTUM-SAFE AND CLASSICAL CRYPTOGRAPHIC SYSTEMS

ISO 9001:2015

दूरसंचार अभियांत्रिकी केंद्र,
खुर्शीदल भवन, जनपथ, नई दिल्ली-110001, भारत
TELECOMMUNICATION ENGINEERING CENTRE
KHURSHIDLAL BHAWAN, JANPATH, NEW DELHI–110001, INDIA
www.tec.gov.in

Release 2.0, _____2026

FOREWORD

Telecommunication Engineering Centre (TEC) functions under the Department of Telecommunications (DoT), Government of India. Its activities include:

- Issue of Generic Requirements (GR), Interface Requirements (IR), Service Requirements (SR) and Standards for Telecom Products and Services
- Field evaluation of products and Systems
- National Fundamental Plans
- Support to DoT on technology issues
- Testing & Certification of Telecom products

For testing, four Regional Telecom Engineering Centres (RTECs) have been established, which are located in New Delhi, Bangalore, Mumbai, and Kolkata.

ABSTRACT

Cryptographic systems are essential for securing communication and protecting sensitive data from unauthorized access. This document describes the generic requirements and specifications of Quantum-safe and Classical Cryptographic systems. This document specifies the generic requirements and technical specifications for Classical and Quantum-Safe Cryptographic Systems, including cryptographic modules and their secure implementation, operation, and management. The standard aims to provide a unified specification to support secure communication, data protection, and cryptographic agility in the presence of evolving classical and quantum threats.

The document further specifies technical specification requirements for products and services offered by vendors for the purpose of testing, certification, and compliance assessment under desirable requirements. In addition, guidelines for procurers are provided to support informed procurement, deployment, operation, and maintenance of cryptographic systems in a secure and interoperable manner.

This standard enables indigenous certification aligned with global practice, avoids

vendor lock-in and provides a future-ready specification for classical, hybrid, and quantum-safe cryptographic systems to supports quantum-safe transition in, Digital Public Infrastructure, Government and critical infrastructure.

Table of Contents

HISTORY SHEET

S. No.	GR No.	Title	Remarks
1.	TEC 91010 : 2023	Generic Requirements of Quantum-safe and Classical Cryptographic Systems	First issue
1.	TEC 91010: 2026 (Rev 1.0)	Generic Requirements of Quantum-safe and Classical Cryptographic Systems	Revision 1.0

REFERENCES

The following referenced documents are necessary for the application of the present document.

Sr. No.	Document No.	Title/Document Name
1.	CISPR 32/ or IS/CISPR 32: 2015	Limits and methods of measurement of radio disturbance characteristics of Information Technology Equipment
2.	ETSI TR 103 619 V1.1.1 (2020-07)	Migration strategies and recommendations to Quantum-safe schemes
3.	ETSI GS QKD 014 V1.1.1 (2019-02)	Quantum Key Distribution (QKD); Protocol and data format of REST-based key delivery API
4.	FIPS 140-3	Security Requirements for Cryptographic Modules
5.	FIPS PUB 197	Advanced Encryption Standard (AES) 2001
6.	FIPS PUB 198	The Keyed-Hash Message Authentication Code (HMAC) 2002
7.	IEC 60825-2/ IS 14624-2	Safety of laser products Part 2 safety of optical fibre communication systems OFCS (First Revision)
8.	IEC 61000-4-11/ IS 14700 (Part 4/Sec 11):2020	Testing & measurement technique- voltage dips, short interruptions, and voltage variations immunity tests.
9.	IEC 61000-4-2 / IS 14700 (Part 4/Sec 2): 2018	Testing and measurement techniques of Electrostatic discharge immunity test
10.	IEC 61000-4-29	Testing and measurement techniques- Voltage dips, short interruptions, and voltage variations on D.C input power port immunity test.

11.	IEC 61000-4-3/. IS 14700 (Part 4/Sec 3): 2010	Radiated RF electromagnetic field immunity test
12.	IEC 61000-4-4/ IS 14700 (Part 4/Sec 4): 2018	Testing and measurement techniques of electrical fast transients/burst immunity test
13.	IEC 61000-4-5(2017)/ IS 14700 (Part 4/Sec 5): 2019	Testing & Measurement techniques for surge immunity test.
14.	IEC 61000-4-6 / IS 14700 (Part 4/Sec 6): 2016	Testing & Measurement techniques for surge immunity test and Immunity to conducted disturbances
15.	IEEE 802.1AE	Media Access Control (MAC) Security
16.	IEEESTD.2018.8585421	IEEE Standard for Local and metropolitan area networks–Media Access Control (MAC) Security. IEEE. December 2018.
17.	IEC 60215/ IS 10437(1986)	Safety requirements for radio transmitting equipment
18.	IEC 60950-1(2005)/ IS 13252 (2010)	Safety of information technology equipment
19.	ISO/IEC 10116:2006 / IS 15116 : 2018	Information technology – Security techniques – Modes of operation for an n-bit block cipher
20.	ISO/IEC 18033-3:2010/	Information Technology Security Techniques Encryption algorithms
21.	ISO/IEC 19790:2025	*Information security, cybersecurity and privacy protection — Security requirements for cryptographic modules*
22.	ISO/IEC 24759:2025	*Information security, cybersecurity and privacy protection — Test requirements for cryptographic modules*

23.	ITU-T X.1710	Security framework for quantum key distribution networks Series X: Data Networks, Open System Communications And Security
24.	ITU-T X.1810	Framework for quantum-safe cryptography
25.	ITU-T X.1811	Security guidelines for applying quantum-safe algorithms in IMT-2020 systems
26.	ITU-T X.1812	Quantum-safe cryptographic mechanisms and applications
27.	ITU-T X.800	Security architecture for Open Systems Interconnection for CCITT applications
28.	ITU-T Y.3802	Quantum key distribution networks - Functional architecture
29.	ITU-T Y.3803	Quantum key distribution networks - Key management
30.	ITU-T Y.3804	Quantum key distribution networks - Control and management
31.	NISTIR 8105	Report on Post-Quantum Cryptography
32.	TEC/SD/DD/EMC-221/05/OCT-16	Electromagnetic Compatibility Standard for Telecommunication Equipment
33.	QM-333	Specification for environmental testing of electronic equipment for transmission and switching use
34.	RFC 3602	The AES-CBC Cipher Algorithm and Its Use with IPsec
35.	RFC 3686	Using Advanced Encryption Standard (AES) Counter Mode with IPsec Encapsulating Security Payload (ESP)
36.	RFC 4106	The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating
37.	RFC 4301	Security Architecture for the Internet Protocol
38.	RFC 4302	IP Authentication Header
39.	RFC 4303	IP Encapsulating Security Payload

40.	RFC 4307	Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)
41.	RFC 4308	Cryptographic Suites for IPsec
42.	RFC 4868	Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec
43.	RFC 5282	Using Authenticated Encryption Algorithms with the Encrypted Payload of the Internet Key Exchange Protocol Version 2 (IKEv2)
44.	RFC 7296	Internet Key Exchange Protocol Version 2 (IKEv2)
45.	RFC 7321	Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)
46.	RFC 8446	TLS 1.3, the current baseline secure transport protocol and the primary IETF vehicle for post-quantum (PQC) and hybrid cryptography deployment.
47.	RFC 9052	Authoritative specification for CBOR Object Signing and Encryption (COSE) for signing, encryption, and authentication of data objects
48.	RFC 9180	Hybrid Public Key Encryption (HPKE), a modern, flexible framework for public-key encryption using KEMs, designed to support hybrid and post-quantum–ready deployments.
49.	RFC 7696	Guidelines for Cryptographic Key Management
50.	RFC 8221	Mandatory Algorithms for DNSSEC
51.	RFC 5751	Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification
52.	RFC 4880	OpenPGP Message Format
53.	RFC 5280	Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile

54.	ISO/IEC 15408 (All Parts)	Common Criteria for Information Technology Security Evaluation

Note: Unless otherwise explicitly stated, the latest approved issue version of the standards/documents referred to above, with all amendments in force, on the issuance date of this GR shall be applicable.

CHAPTER-1

Introduction to Cryptographic Systems

1.1. Introduction to Cryptographic systems

Cryptography is the practice of securing communication and protecting data from unauthorized access by converting plaintext into ciphertext using mathematical algorithms, making it unintelligible to anyone without the proper key. It plays a critical role in securing our digital infrastructure.

The typical cryptographic system is shown in Figure 1. The original message is usually termed plaintext and the scrambled message is called the ciphertext. The encryption algorithm converts the plaintext to the ciphertext and the decryption algorithm performs a reverse process to get back the original message.

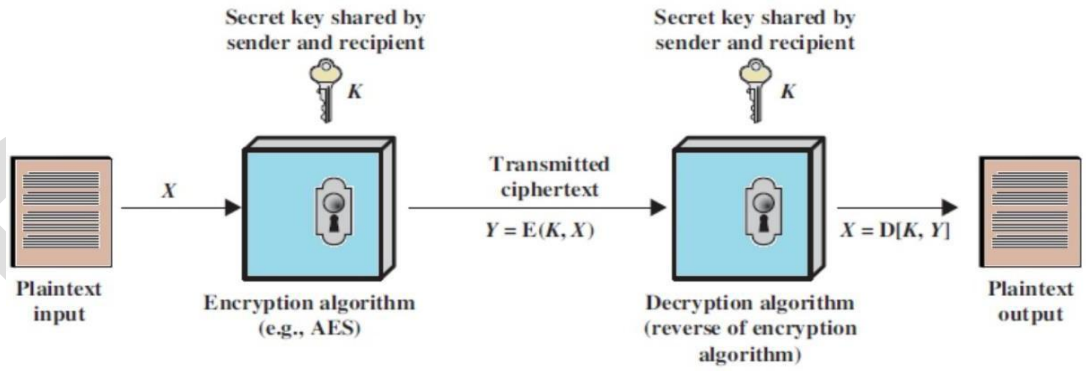

Figure 1: Block Diagram of a typical Cryptographic System

Our most crucial communication protocols rely on three core cryptographic primitives: *public key encryption*, *digital signatures* and *key exchange*. These primitives are implemented using state-of-the-art of cryptographic algorithms, e.g., AES, Diffie-Hellman Key Exchange(DHKE), the RSA (Rivest-Shamir-Adleman) algorithm, and elliptic curve cryptography(ECC).

The security of the public key cryptographic primitives as mentioned above depends on the difficulty of a number of theoretical problems, such as Integer Factorisation and the Discrete Log problem. In 1994, Peter Shor showed that

Quantum computers, a new technology leveraging the physical properties of matter and energy to perform calculations, can efficiently solve factorisation and discrete log problems, thereby rendering all public key cryptosystems based on such assumptions insecure. Thus, a sufficiently powerful quantum computer will peril many forms of modern communication, from Key exchange to encryption to digital authentication. As a result, RSA and DHKE are no longer secure in a post-quantum era.

Further, for data encryption, symmetric algorithms such as AES are widely used. Grover's algorithm offers a quadratic speed-up for brute-force key search compared to classical search, therefore, it can affect AES (and other symmetric encryption algorithms) by reducing the effective security of an n-bit key n/2 bit security on a sufficiently powerful quantum computer—so, for example, AES-128 would offer roughly ~64-bit quantum security, which is why doubling the symmetric key length is generally considered necessary to maintain the same security margin against quantum adversaries but not solely sufficient to prevent attack from quantum computer.

Table 1: Impact of Quantum Computing on common cryptographic algorithms

Sl. No.	Cryptographic Algorithms	Type	Purpose	Impact of the large scale quantum computer
1	AES	Symmetric Key	Encryption	Larger key sizes needed
2	SHA-2, SHA-3	----------	Hash functions	Larger output needed
3	RSA	Public key	Signatures, key establishment	No longer secure
4	ECDSA, ECDH	Public key	Signatures, key exchange	No longer secure

Quantum-safe cryptographic systems, also known as post-quantum cryptography, are designed to be resistant to attacks from both classical and

quantum computers. These systems use algorithms that are believed to be secure even against quantum computers. Quantum-safe cryptography is becoming increasingly important as quantum computers continue to evolve and become more powerful.

It is, therefore, critical to begin planning the replacement of hardware, software, and services that can interoperate with existing communications protocols and networks. Most quantum-resistant algorithms have larger Key sizes than the ones they will substitute, which is a big challenge. Quantum-safe algorithms may change various Internet protocols, such as the Transport Layer Security (TLS) protocol or the Internet Key Exchange (IKE). Implementing quantum-safe algorithms requires identifying hardware and software modules, operating systems, communication protocols, cryptographic libraries, and applications employed in data centres on-premises or in the cloud and distributed computing, storage, and network infrastructures. From a compliance and risk-management perspective, transitioning to quantum-safe cryptography helps address long-term confidentiality concerns such as "harvest now, decrypt later," where encrypted sensitive data captured today could potentially be decrypted in the future if cryptographically relevant quantum capabilities emerge.

1.2. Classification of cryptographic algorithms

Cryptographic algorithms are broadly classified into two categories, traditional and modern, based on the type used during the encryption and decryption process (refer to figure 2).

1.2.1. Traditional cryptography

Traditional cryptography refers to cryptographic methods and techniques developed before the advent of computers.

1.2.2. Modern Cryptography

Modern cryptography is based on publicly known mathematical algorithms that operate on binary bit sequences and utilise secret keys. There are three types of modern cryptography:

i Symmetric (Secret Key) cryptography

ii Asymmetric (Public Key) cryptography

iii Cryptographic Hash Functions

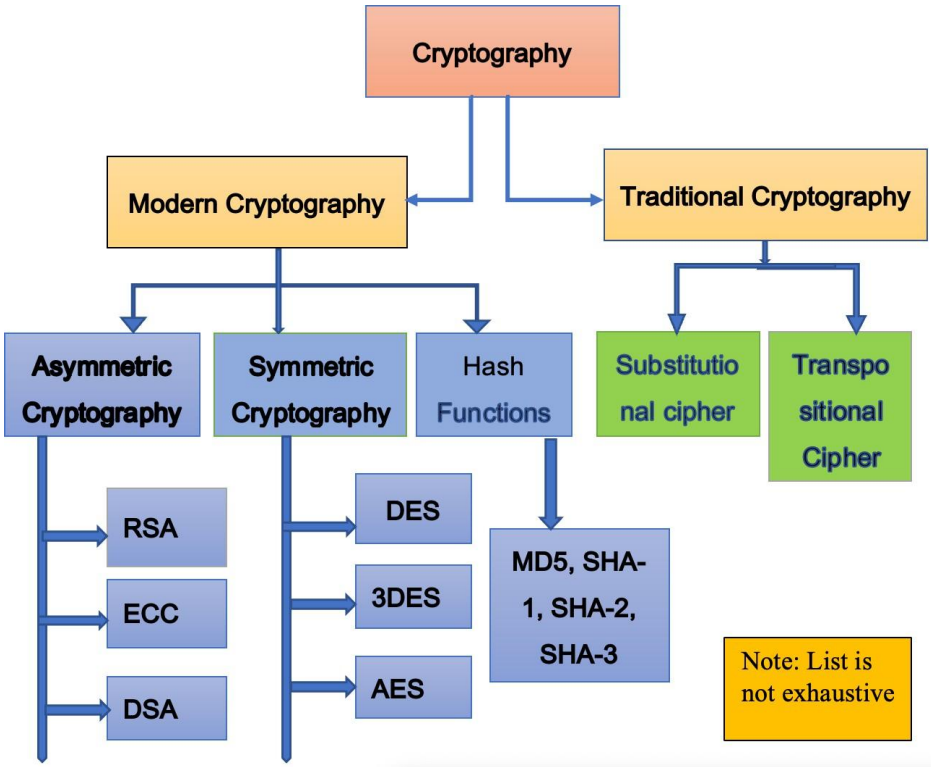

Figure 2: Block Diagram of classification of classical cryptography

1.2.2.1 Symmetric key cryptography

Encryption and decryption keys are identical in this scheme and should be known only to the communicating parties. Symmetric key cryptography is much faster than Asymmetric key cryptography, is far less resource-intensive than asymmetric encryption and is an incredibly efficient way to protect large volumes of data. Examples are Advanced Triple-Data Encryption Standard (DES), i.e., 3DES, Advanced Encryption System (AES), etc.

1.2.2.2 Asymmetric key cryptography

In this scheme, two keys are used, i.e., public key (for encryption) and private key (for decryption). The private key is kept secret as it is used for decryption, while the public key is not. For a secure public key cryptosystem, it is impossible to determine the private key's value by knowing the corresponding public key.

Most public communication networks use a combination of asymmetric and symmetric key cryptography schemes. An asymmetric/ Public Key

Cryptography scheme is used for key distribution. At the same time, the data flow is secured using a symmetric technique because of its better performance in the encryption/decryption process.

1.2.2.3 Hash Function

A Hash function is a cryptographic algorithm that takes an input message of any size and outputs a short fingerprint of fixed length. Typically, it does not require any key along with the input message, and the output is usually called hash-value or hash-digest. These algorithms are typically used to ensure the authenticity or integrity of data. Hash functions can also use keys, referred to as Keyed-hash functions, under such usage. Many operating systems/applications store passwords using hash functions.

1.3. Types of configuration of cryptographic system

1.3.1.1 A cryptographic module shall be a set of hardware, software, firmware or some combination thereof that at a minimum, implements a defined cryptographic service employing an approved cryptographic algorithm, security function or process and contained within a defined cryptographic boundary.

1.3.1.2 The cryptographic systems can be classified based on the hardware, software and or firmware used in modular form within the cryptogpraphic boundary. These modules may be part of any interdependent or standalone system.

The cryptographic module/system can be defined as one of the following types:

i. Hardware module: It is a module whose cryptographic boundary is specified at a hardware perimeter. Firmware and/or software, which may also include an operating system, may be included within the hardware cryptographic boundary.

ii. Software module: It is a module whose cryptographic boundary delimits the exclusive software component(s) (may be one or multiple software components) that execute(s) in an adjustable operational environment. The computing platform and operating system of the working environment in which the software performs are external to the defined software module

boundary.

iii. Firmware module: It is a module whose cryptographic boundary delimits the exclusive firmware component(s) that execute(s) in a limited or non-modifiable operational environment. The computing platform and operating system of the operational environment in which the firmware executes in are external to the defined firmware module boundary but explicitly bound to the firmware module.

iv. Hybrid Software module: It is a module whose cryptographic boundary delimits the composite of a software component and a disjoint hardware component (i.e. the software component is not contained within the hardware module boundary). The computing platform and operating system of the operational environment in which the software executes are external to the defined hybrid software module boundary.

v. Hybrid Firmware module: It is a module whose cryptographic boundary delimits the composite of a firmware component and a disjoint hardware component (i.e. the firmware component is not contained within the hardware module boundary). The computing platform and operating system of the operational environment in which the firmware executes in are external to the defined hybrid firmware module boundary but explicitly bound to the hybrid firmware module.

1.3.1.3 Classification of Quantum-safe cryptography configuration

The Quantum-safe cryptography module can be classified in a similar manner to classical cryptography modules. However, the algorithms will be different, especially for public key infrastructure like public key encryption schemes, key exchange mechanisms, digital signature schemes and hash functions. These algorithms need to resist attacks by quantum computers, and at the same time, they should still be secure against classical computer attacks.

For symmetric key cryptography, doubling the key size can provide some protection against quantum computing attacks, but this is not a complete solution. New search algorithms are being developed for asymmetric key cryptography to resist quantum computing attacks.

1.3.1.4 Quantum-safe symmetric cryptography

Symmetric key cryptography is vulnerable to quantum attacks. It is mostrly threatened by Grover's algorithm. Unlike the asymmetric encryption algorithms (eg. RSA, etc) which could be completely breaken by the Quantum computer; for symmetric algorithms like AES, the best known Gorver's algorithm for attacking these encryption algorithms only weakens them. Grover's algorithm decreases the effective key length of a symmetric encryption algorithm by half, so AES-128 has an effective key space of 2^64 and AES-256 has an effective key space of 2^128.

1.3.1.5 Quantum-safe asymmetric cryptography

Today's most important uses of public key cryptography are for digital signatures and key establishment. Constructing a large-scale quantum computer would render many of these public key cryptosystems insecure. In particular, this includes those based on the difficulty of integer factorisation, such as RSA and those based on the hardness of the discrete logarithm problems. Quantum-safe Cryptography mainly refers to developing new asymmetric cryptography techniques that use a different class of hard mathematical problems. There are a few popular Quantum-safe cryptographic approaches that have emerged, such as Lattice-based, Code-based, multivariate-based and hash based cryptography. These mathematically hard problems are believed to be secure against classical as well as quantum computers.

1.3.1.6 Quantum-safe Hash and Signature functions

Cryptographic hash functions are widely used to provide data integrity and to build digital signature schemes. SHA-2 family hash algorithms (e.g., SHA-256 and SHA-512) are generally considered quantum-safe in the sense that no known quantum algorithm (including Shor's) breaks them outright; instead, the main quantum impact is limited to generic speed-ups such as Grover's algorithm. While Grover's algorithm can reduce the complexity of brute-force search, this impact can typically be addressed by selecting appropriately strong hash functions and security parameters.

Hash-based signature schemes are an important class of quantum-safe digital signatures. Basic constructions such as Lamport–Diffie and Winternitz are one-time signature schemes, meaning each private signing key must be used only once. To enable practical use for multiple signatures, these one-time keys are combined using Merkle tree constructions, allowing a single public key to authenticate a large number of signatures, bounded by the size of the tree. A well-known example is the eXtended Merkle Signature Scheme (XMSS), which is a stateful hash-based signature scheme; it requires careful state management to ensure that one-time keys are never reused. Overall, quantum-safe hash and signature mechanisms provide a robust alternative for digital signatures in a post-quantum transition, particularly for long-term integrity and authenticity requirements.

1.4. Elements or Subsystems and Applications of a cryptographic systems

A cryptographic system relies upon two basic components, i.e., an algorithm (or cryptographic methodology) and a cryptography key. Cryptographic subsystems in classical cryptography are the same as in Quantum-safe cryptographic systems except that different algorithms are implemented on hardware (Key sharing methods are different in Quantum Key Distribution (QKD) and Quantum-safe Cryptography). It also consists of software/firmware modules, operating systems, communication protocols, cryptography libraries, and applications deployed in data centres on-premises or in cloud, distributed computing, storage and network infrastructure.

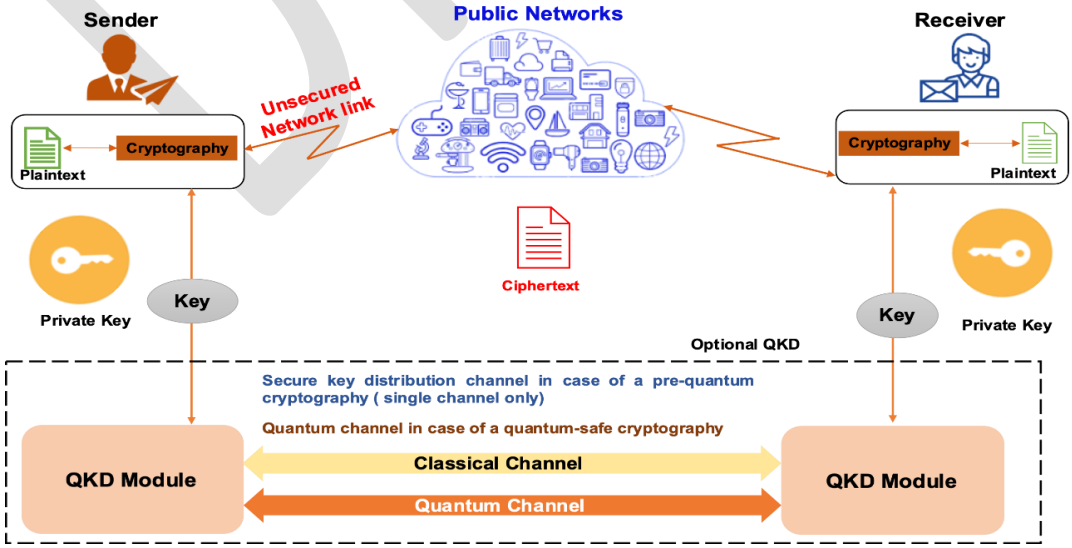

Figure 3: Block Diagram of a Symmetric cryptographic system

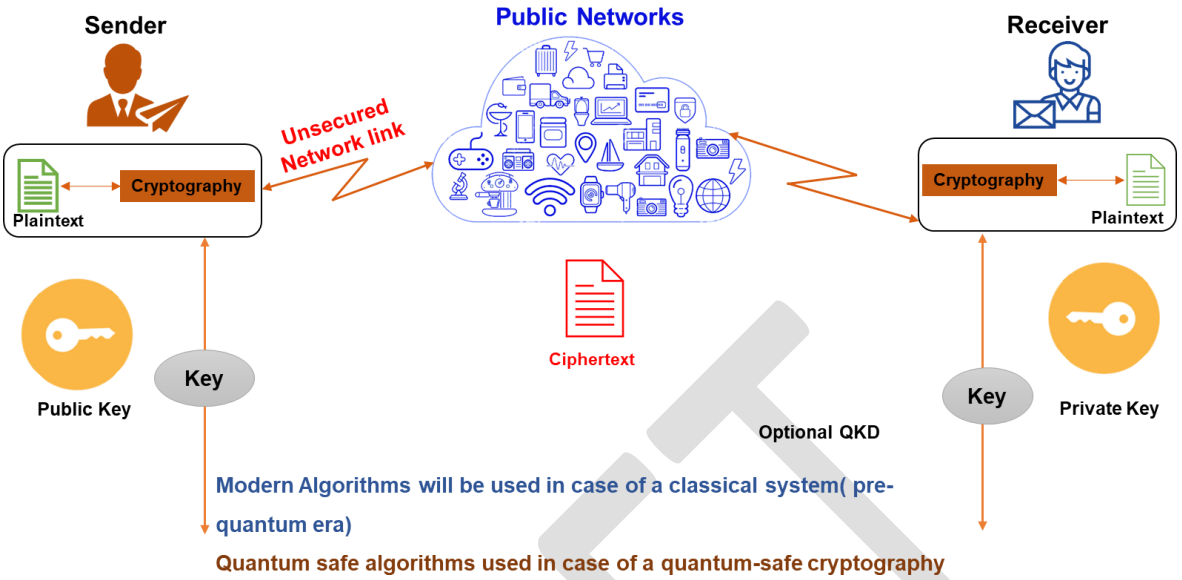

Figure 4: Block Diagram of Asymmetric cryptographic system

Note: Encryption algorithms are the same, but in symmetric cryptographic systems, the key is transported through quantum modules over the QKD channel, whereas in the case of Asymmetric cryptography systems, the key is shared using Quantum-safe Cryptography key sharing algorithms. QKD is one of the key sources, as shown in Figure 3.

CHAPTER-2

Functional Requirements

2.1. Elements or Subsystems of Cryptographic systems

2.1.1. Encryptor/Decryptor

Encryptor communicates data over an unsecured network by changing it from plain text to cipher text using an encryption algorithm driven by Key. The Decryptor at receiver, who holds the same key and decryption algorithm, turns the cipher text into plain text. In this way, data transmit securely over an unsecured communication channel

Table 2- Symmetric Key Encryptor/Decryptor Requirements

Sl. No.	Requirement Title	Requirement Description
1	Correctness of Encryption/Decryption	The encryption module shall ensure that data encrypted using a supported algorithm and subsequently decrypted using the corresponding key and parameters results in the original plaintext without loss or modification.
2	Key Validation	The encryption module shall support cryptographic keys of valid lengths (e.g., 128-bit, 192-bit, and 256-bit, as applicable) and shall reject keys of invalid or unsupported lengths.
3	Cipher Mode Behavior	The encryption module shall correctly implement supported block cipher modes, including but not limited to ECB, CBC, CFB, OFB, CTR, and GCM, in accordance with the applicable cryptographic specifications.
4	IV Handling	The encryption module shall generate and use Initialization Vectors (IVs) correctly, ensuring sufficient randomness for

		applicable modes (e.g., CBC, CTR), and shall correctly apply the IV during decryption.
5	Input Validation	The encryption module shall handle edge-case inputs, including empty, null, or oversized plaintexts, in a controlled and secure manner without causing unexpected behavior or security compromise.
6	Ciphertext Integrity	The encryption module shall detect modification or tampering of ciphertext and shall fail decryption or integrity verification when such tampering is detected.
7	Determinism vs. Non-Determinism	For non-deterministic encryption modes, the encryption module shall ensure that encrypting the same plaintext with the same key but different IVs results in different ciphertexts.
8	Interoperability	The encryption module shall ensure interoperability such that ciphertext generated by one compliant implementation can be successfully decrypted by another implementation conforming to the same cryptographic standard and parameters.
9	Padding Validation	Where padding is applicable, the encryption module shall correctly implement standard padding schemes (e.g., PKCS#7) and shall generate an error when incorrect or invalid padding is encountered during decryption.
10	Secure Key Disposal	The encryption module shall securely clear cryptographic keys and sensitive intermediate data from memory immediately after use to prevent residual data exposure.

11	Performance Requirements	The encryption module shall perform encryption and decryption operations within acceptable time limits under defined operating conditions, ensuring suitability for the intended deployment environment.

Table 3- Asymmetric Key Encryptor/Decryptor Requirements

Sl. No.	Requirement Category	Requirement Description	Applies To
1	Key Generation	The cryptographic module shall generate valid key pairs in accordance with the applicable algorithm specifications, ensure correct key format, and guarantee randomness and uniqueness of generated keys.	RSA, ECDH, ECC, KEM
2	Encryption / Decryption	The cryptographic module shall correctly encrypt plaintext using the appropriate public key and shall correctly decrypt the resulting ciphertext using the corresponding private key, such that the decrypted output matches the original plaintext. The module shall validate padding schemes and generate an error upon detection of corrupted or invalid ciphertext.	RSA, ECC
3	Digital Signature / Signature Verification	The cryptographic module shall support digital signature generation using a private key and signature verification using the corresponding public key. The module shall detect and reject signatures generated over tampered data and shall support secure hash algorithms such as	RSA, ECC

		SHA-2 and SHA-3.	
4	Key Agreement	The cryptographic module shall support secure key agreement mechanisms by generating ephemeral key pairs for two parties and deriving a shared secret. The module shall verify key agreement parameters and shall reject invalid or weak parameters.	DH, ECDH
5	Key Encapsulation Mechanism (KEM)	The cryptographic module shall support key encapsulation using a recipient's public key and shall correctly decapsulate the encapsulated key using the corresponding private key. The module shall ensure correctness of the recovered key and shall handle corrupted encapsulated data securely.	KEM (e.g., ML-KEM, Hybrid schemes)
6	Error Handling and Boundary Conditions	The cryptographic module shall reject invalid keys, malformed inputs, and unsupported formats. The module shall generate appropriate errors for boundary conditions, including oversized inputs, invalid structures, and message size violations, without exposing sensitive information.	All
7	Performance and Timing	The cryptographic module shall operate within defined performance limits for key generation, encryption, decryption, and signature operations. The module shall be designed to mitigate timing attacks and side-channel leakage.	All
8	Interoperability	The cryptographic module shall ensure interoperability with other compliant	All

		implementations. Keys, ciphertexts, signatures, and encapsulated data generated by the module shall be usable and verifiable across conforming platforms implementing the same standards.	

2.1.2. Hash Functions-

Hashing is a method used to verify data integrity (already referred to in para 1.2.2.3). This technique is referred to as collision resistance, refer to figure 5.

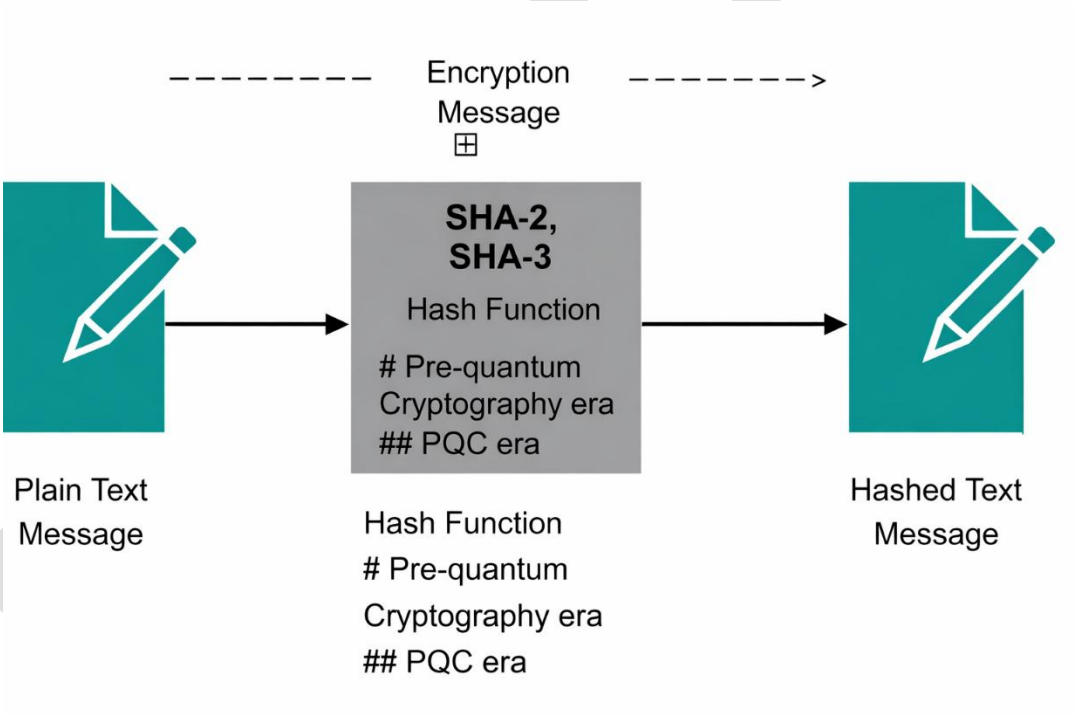

Figure 5: Block Diagram of Hash functions

i) A Message Digest 5 algorithm [MD5]: This creates a 128-bit digest used in the hash function. (Not recommended for use).

ii) Secure Hash Algorithm 1 (SHA-1): This creates a 160-bit digest (Not recommended for use).

iii) Secure Hash Algorithm 2 (SHA-2): Options include a digest between 224 and 512 bits.

iv) Secure Hash Algorithm 3 (SHA-3): Options include a digest between 224

and 512 bits.

Table 4- Hash Algorithm Requirements

Sl. No.	Requirement Category	Requirement Description
1	Empty Input Handling	The hash function shall correctly process an empty input string and shall produce a hash output that exactly matches the official reference digest for empty input as defined in the applicable standard.
2	Known Answer Tests (KAT)	The hash function shall correctly process known fixed input values and shall produce outputs that exactly match the official test vectors defined in the applicable cryptographic standard.
3	Variable Input Length Handling	The hash function shall correctly process inputs of varying lengths, including but not limited to 1 byte, 10 bytes, 1 KB, and 1 MB, and shall produce outputs that match the expected reference hashes for each input length.
4	Block Boundary Handling	The hash function shall correctly process input sizes around the internal block boundary (e.g., 136 bytes for SHA3-256), apply padding correctly, and shall produce outputs that match the applicable reference vectors.
5	Incremental Hashing	The hash function shall support incremental hashing by processing input data in multiple chunks, and shall ensure that the resulting hash output is identical to the hash computed over the same data processed in a single pass.

6	Collision Resistance Validation	The hash function shall ensure that distinct input values produce distinct hash outputs under test conditions, with no collisions observed during validation testing.
7	Avalanche Property	The hash function shall exhibit the avalanche effect such that a one-bit change in the input results in a significant change in the output, with approximately 50% of output bits differing.
8	Extendable Output Function (XOF) Support	Where SHAKE or other extendable-output variants are supported, the hash function shall generate outputs of arbitrary requested length and shall ensure that the output length exactly matches the requested length.
9	Padding Verification	The hash function shall correctly implement the multi-rate padding scheme defined for SHA-3 and related functions, and shall produce hash outputs that match the applicable reference vectors.
10	Domain Separation	The hash function shall ensure domain separation such that different SHA-3 variants produce distinct outputs for the same input, with no overlap or ambiguity between outputs.
11	Performance and Stress Handling	The hash function shall correctly process large input sizes under stress conditions and shall complete hashing operations within defined acceptable performance limits.
12	Standards Compliance	The hash function shall demonstrate full compliance with the applicable cryptographic standard by successfully passing all official test vectors without deviation.

Note – SHA-1 has been deprecated so shall not be used.

2.1.3. Hashed Message Authentication Code (HMAC)

Instead of using a hash that anyone can calculate, it includes a secret key. Currently, there are three approved general purpose MAC algorithms: HMAC, KMAC and CMAC.

Table 5- HMAC Test Requirements

Sl. No.	Requirement Category	Requirement Description
1	Key Handling	The HMAC module shall correctly process keys of all valid lengths, including empty keys, keys shorter than the underlying hash block size, keys equal to the block size, and keys longer than the block size. For keys longer than the block size, the module shall hash the key prior to HMAC computation and shall produce outputs that match the applicable official test vectors.
2	Message Length Handling	The HMAC module shall correctly compute HMAC values for messages of varying lengths, including empty messages, short messages, and long multi-block messages, and shall produce outputs that match the applicable test vectors.
3	Known Answer Tests (KAT)	The HMAC module shall correctly compute HMAC values for known key/message pairs and shall generate outputs that exactly match the official reference test vectors defined in the applicable standard.

4	Incremental Processing	The HMAC module shall support incremental processing of input data split into multiple chunks and shall ensure that the resulting HMAC value is identical to the HMAC computed over the same data processed in a single operation.
5	Output Length	The HMAC module shall ensure that the HMAC output length exactly matches the output length of the underlying hash function used for HMAC computation.
6	Security Properties	The HMAC module shall ensure that small changes in the key or message result in significantly different HMAC outputs, demonstrating the avalanche property and resistance to trivial forgery.
7	Hash Function Variants	The HMAC module shall support HMAC computation using approved hash functions, including SHA-224, SHA-256, SHA-384, and SHA-512, and shall produce correct outputs for each supported hash variant.
8	Performance	The HMAC module shall compute HMAC values for very large messages within defined acceptable performance and timing limits, suitable for the intended deployment environment.

2.1.4. Random Number Generator

In cryptography, randomness is found everywhere, from the generation of keys to encryption systems, even how cryptosystems are attacked. Without randomness, all crypto operations would be predictable and hence, insecure. A good random number generator consists of two parts: a source of entropy and

a cryptographic algorithm. Cryptographic algorithms require Keys. A Random Number Generator (RNG), also called a Random Bit Generator (RBG), is needed in the key generation process to create a random (strong) key as well as for other cryptographic purposes such as initialisation vectors and nonces. Typically, a True Random Number Generator (TRNG) provides a source of randomness or "entropy" to seed a Pseudo-Random Number Generation (PRNG), also called a Deterministic Random Bit Generator (DRBG).

Random bit generation	ISO/IEC 18031 TEC GR QRNG TEC 91020:2024 Test Guide of GR QRNG TEC 91021:2024 NIST SP 800-90 Series	Developers must demonstrate that their entropy source is sufficiently random through a combination of design and/or test processes and continuous checks during operation for any fault that could have catastrophic consequences for generating secure cryptographic keys.

2.1.5. Digital Signatures

Offers Authentication, Data Integrity, and Non-repudiation. Digital signatures involve public and private key pairs, hashing, and encryption.

Table 6- Signature Test Requirements

Sl. No.	Requirement Category	Requirement Description
1	Signature Generation	The digital signature module shall successfully generate signatures for messages of varying sizes and content, including empty messages, and shall produce signatures that conform to the expected format defined by the applicable algorithm specification.

2	Signature Verification	The digital signature module shall correctly verify valid signatures generated using the corresponding public key and shall reject signatures when verified with incorrect keys or when the signed message has been altered. The module shall support verification of messages of varying lengths.
3	Known Answer Tests (KAT)	The digital signature module shall correctly perform signature generation and verification for known key/message/signature tuples and shall produce results that exactly match the official test vectors defined in the applicable standard.
4	Boundary and Edge Case Handling	The digital signature module shall correctly sign and verify messages at boundary sizes, including zero-length messages and the maximum supported message size, without errors. The module shall enforce minimum and maximum supported key size constraints as defined by the applicable algorithm.
5	Security Properties	The digital signature module shall provide resistance to signature forgery and replay attacks, ensuring that each signature is cryptographically unique and bound to the specific message and key. The module shall implement appropriate protections against side-channel leakage during signature generation, as applicable to the implementation environment.
6	Algorithm Interoperability	The digital signature module shall ensure interoperability such that signatures generated by one compliant implementation can be successfully verified by other independent implementations conforming to the same algorithm specification.

7	Performance	The digital signature module shall generate and verify signatures within defined acceptable performance limits for supported key sizes and message lengths, suitable for the intended deployment environment.
8	Post-Quantum Signature Support	Where post-quantum algorithms are supported, the digital signature module shall correctly implement key generation, signature generation, and signature verification for approved post-quantum algorithms (e.g., ML-DSA, FN-DSA) and shall conform to their respective algorithm specifications and test vectors.

2.1.6. Key Management

Deals with generating keys, verifying keys, exchanging keys, storing keys, and at the end of their lifetime, destroying keys. The bigger the key, the more secure the algorithm will be. The only negative of having an extremely long key is that the longer the key, the more the CPU is used to decrypt and encrypt data.

Table 7 - List of the functional tests for Key Management

Sl. No.	Requirement Category	Requirement Description
1	Key Generation	The key management module shall generate symmetric cryptographic keys (e.g., AES) of supported lengths using approved random number generation mechanisms, ensuring sufficient entropy and uniform randomness. The module shall generate asymmetric key pairs (e.g., RSA, ECC) that meet the required key size, format, and parameter specifications. Where post-quantum cryptography is supported, the module shall generate quantum-safe key pairs (e.g., ML-KEM, NTRU, FN-DSA) in compliance with the applicable PQC algorithm specifications.

2	Key Storage and Protection	The key management module shall securely store cryptographic keys in protected storage and shall ensure confidentiality and integrity of stored keys. The module shall support secure key wrapping and unwrapping mechanisms (e.g., AES Key Wrap) and shall ensure that wrapped and unwrapped keys remain intact and unaltered.
3	Key Distribution and Transport	The key management module shall support secure key exchange and distribution mechanisms, including Diffie–Hellman, ECDH, and post-quantum key encapsulation mechanisms (KEMs). The module shall ensure that keys exchanged via in-band or out-of-band transport methods are delivered securely without disclosure, modification, or unauthorized access.
4	Key Usage and Lifecycle Management	The key management module shall ensure that cryptographic keys can be correctly used for their intended purposes, including encryption, decryption, digital signature generation, and verification. The module shall enforce key lifecycle states, including key activation, suspension, revocation, expiration, and secure destruction, in accordance with defined key management policies.
5	Key Backup and Recovery	The key management module shall support secure backup and recovery of cryptographic keys while preserving confidentiality and integrity. The module shall ensure that recovered keys are functionally equivalent to the original keys and remain protected against unauthorized access.
6	Known Answer Tests (KAT)	The key management module shall validate key-related operations against official known answer test vectors and shall produce results that exactly match the reference outputs defined in the

		applicable standards.
7	Interoperability	The key management module shall support secure import, export, and use of cryptographic keys across different compliant systems and platforms, ensuring interoperability without loss of security or functionality.
8	Performance	The key management module shall perform key generation, wrapping, unwrapping, backup, recovery, and exchange operations within defined acceptable performance limits suitable for the intended operational environment.
9	Security Properties	The key management module shall provide resistance against key extraction attempts and shall implement appropriate protections against side-channel attacks during key generation, storage, distribution, and usage. No unauthorized leakage of key material shall occur under normal or stress operating conditions.

2.1.7. Key Management Interoperability Protocol (KMIP)

Deals with generating keys, KMIP protocol allows communication between key management systems and cryptographically enabled applications, such as email, databases, and storage devices. KMIP is an extensible communication protocol for manipulating cryptographic keys on a key management server that defines message formats. Clients can also ask a server to encrypt or decrypt data without directly accessing the key using KMIP. The key management interoperability standard can support legacy systems and quantum-safe cryptographic applications.

Table 8 - List of the functional tests for KMIP

Sl. No.	Requirement Category	Requirement Description

1	Key Lifecycle Management	The key management system shall support creation of cryptographic keys, including symmetric, asymmetric, and post-quantum cryptography (PQC) keys, with correct type, attributes, and metadata. The system shall support key state transitions including activation, deactivation, revocation, and shall ensure that revoked keys are marked unusable. The system shall securely destroy keys such that the key material and all associated metadata are permanently deleted.
2	Key Query and Retrieval	The key management system shall allow retrieval of keys using a unique identifier. The system shall support locating keys based on attributes and shall return accurate key metadata, including algorithm, key length, permitted usage, and lifecycle state.
3	Key Usage Operations	The key management system shall allow authorized use of keys for cryptographic operations, including encryption and decryption using symmetric or asymmetric keys, digital signature generation and verification using RSA, ECDSA, or supported PQC algorithms (e.g., ML-DSA), and key wrapping and unwrapping using approved mechanisms (e.g., AES or RSA). The system shall enforce correct success or failure behavior based on key type, state, and policy.

4	Key Format and Interoperability	The key management system shall support secure import and export of keys in standard formats, including PEM, DER, and defined PQC binary formats, without corruption or loss of key attributes. The system shall ensure interoperable key handling across different vendor implementations compliant with KMIP.
5	Key Attributes and Policy Enforcement	The key management system shall support assignment of cryptographic usage masks (e.g., encrypt, decrypt, sign, verify) to keys and shall enforce these restrictions during key usage operations. The system shall support configuration of key activation and expiration dates and shall enforce lifecycle timestamps. The system shall allow association of access control policies or tags with keys and shall ensure that such attributes persist and control access as defined.
6	Post-Quantum Compatibility	The key management system shall support registration, storage, and use of post-quantum cryptographic keys, including but not limited to ML-KEM, ML-DSA, and FN-DSA. The system shall support applicable KMIP extensions or profiles required for PQC algorithms and shall ensure that PQC key operations function in accordance with the defined specifications.

7	Error Handling and Security	The key management system shall reject unauthorized key usage attempts and shall return appropriate error responses. The system shall detect and reject corrupted or tampered key material. The system shall deny cryptographic operations using expired or invalid keys in accordance with configured policies.
8	Performance and Scalability	The key management system shall support creation, retrieval, and destruction of a high volume of keys and shall operate within defined performance and scalability thresholds under normal and peak load conditions.

2.1.8. Cryptography Interfaces and APIs

i. **Cryptography API:** Next Generation (CNG) is the long-term replacement for CryptoAPI. CNG is designed to be extensible at many levels and cryptography agnostic in behaviour. CNG is intended for use by developers of applications that will enable users to create and exchange documents and other data in a secure environment, especially over non secure media such as the Internet. At the CNG level, it was necessary to provide substitution and discoverability for all the algorithm types (symmetric, asymmetric, hash functions), random number generation, and other utility functions. The protocol-level changes are more significant because, in many cases, the protocol APIs needed to add algorithm selection and other flexibility options that did not previously exist.

.

Sl. No.	Requirement Category	Requirement Description
1	Algorithm Discovery	The CNG interface shall support dynamic discovery of supported cryptographic algorithms, including symmetric, asymmetric, hash, and random number generation algorithms.
2	Provider Agility	The CNG interface shall allow selection and substitution of cryptographic providers at runtime without requiring recompilation of the application.
3	Cryptographic Correctness	The CNG interface shall correctly perform encryption, decryption, hashing, digital signature generation, and signature verification using approved supported algorithms.
4	RNG Support	The CNG interface shall provide access to a cryptographically secure random number generator suitable for key generation and security-critical operations.
5	Error Handling	The CNG interface shall return appropriate error codes for invalid parameters, unsupported algorithms, and incorrect key usage.
6	Robustness	The CNG interface shall not crash, leak sensitive information, or expose internal cryptographic state under error or misuse conditions.

ii. Web Cryptography API: This specification describes a JavaScript API for performing basic cryptographic operations in web applications, such as hashing, signature generation and verification, and encryption and

decryption. Additionally, it describes an API for applications to generate and/or manage the keying material necessary to perform these operations. Uses for this API range from user or service authentication, document or code signing, and communications' confidentiality and integrity.

Sl. No.	Requirement Category	Requirement Description
1	Secure Context Enforcement	The Web Cryptography API shall be accessible only in a secure execution context and shall prevent cryptographic operations in insecure environments.
2	Cryptographic Operations	The Web Cryptography API shall support hashing, encryption, decryption, digital signature generation, and signature verification using approved algorithms.
3	Key Management	The Web Cryptography API shall support generation and management of cryptographic keys with defined usage and extractability attributes.
4	Key Usage Enforcement	The Web Cryptography API shall enforce key usage restrictions and shall reject unauthorized or disallowed cryptographic operations.
5	Key Protection	The Web Cryptography API shall prevent export of non-extractable keys and shall protect key material from unauthorized access.
6	Random Number Generation	The Web Cryptography API shall provide a secure random number generation function suitable for cryptographic applications.

iii. PKCS #11: This refers to the programming interface to create and manipulate cryptographic tokens (a token where the secret is a cryptographic key). The API defines the most commonly used

cryptographic object types (RSA keys, X.509 certificates, DES/Triple DES keys, etc.) and all the functions needed to use, create/generate, modify and delete those objects. Most commercial certificate authority (CA) software uses PKCS #11 to access the CA signing key or to enroll user certificates.

Sl. No.	Requirement Category	Requirement Description
1	Slot and Token Management	The PKCS #11 interface shall support enumeration of available slots and cryptographic tokens.
2	Session Management	The PKCS #11 interface shall support secure session creation, authentication, role separation, and session termination.
3	Key Generation	The PKCS #11 interface shall support generation of cryptographic keys within the cryptographic token.
4	Key Protection	The PKCS #11 interface shall ensure that sensitive and private keys remain within the cryptographic boundary and are not exportable when marked non-extractable.
5	Cryptographic Operations	The PKCS #11 interface shall support encryption, decryption, signing, and verification using token-resident keys.
6	Access Control	The PKCS #11 interface shall prevent unauthorized access and misuse, including operations without authentication or with unsupported mechanisms.

iv. **Java Cryptography Extension (JCE):** The Java Cryptography Extension (JCE) is an officially released Standard Extension to the Java Platform and part of Java Cryptography Architecture (JCA). JCE provides a framework and implementation for encryption, key generation/management and Message Authentication Code (MAC) algorithms. JCE supplements the

Java platform, which already includes interfaces and implementations of message digests and digital signatures.

Sl. No.	Requirement Category	Requirement Description
1	Provider Architecture	The Java Cryptography Extension shall support provider-based resolution of cryptographic algorithms.
2	Algorithm Agility	The Java Cryptography Extension shall allow dynamic selection and substitution of cryptographic providers without application code changes.
3	Cryptographic Correctness	The Java Cryptography Extension shall correctly perform encryption, decryption, hashing, digital signature generation and verification, and message authentication.
4	Secure Random Generation	The Java Cryptography Extension shall provide access to a cryptographically secure random number generator suitable for key generation.
5	Policy Enforcement	The Java Cryptography Extension shall enforce cryptographic policies such as algorithm restrictions and key size limits.
6	Hardware Integration	The Java Cryptography Extension shall support integration with hardware-backed cryptographic providers where available.

2.1.9. QKD Key delivery interface

The communication protocol is an Application Program Interface (API) that allows authentication and communication between the Cryptographic system by Secure Application Entity (SAE) and the Quantum Key Distribution Entity (QKDE) by Key Management Entity (KME). REST-based APIs are predominantly used due to their simplicity and ease for developers to understand. They are common in many applications; libraries, implementations, and guidance documents are available to the community. Each KME shall have one or multiple QKDEs to connect with other KMEs via QKD links. KMEs shall be able to distribute keys to other KMEs. In each Trusted Node, there shall be at least one KME. One or multiple SAEs may connect with a KME within a Trusted Node, as mentioned in figure 6. It is assumed that each Trusted node is securely operated and managed. Each trusted node shall be located on its site. SAEs shall be located with their connected KMEs on their site. The API between SAE and KME shall be used within a security boundary on each site. KMEs shall provide Web API server functionality to deliver keys to SAEs via HTTPS protocols. Each KME shall have a unique ID (KME ID). A KME ID shall be unique in a QKD network. SAEs make HTTPS requests to KMEs to get keys and status information. Each SAE shall have a unique ID (SAE ID). SAE ID shall be unique in a QKD network.

All communications between SAE and KME shall use the HTTPS protocols (with TLS version 1.3 or higher) (IETF RFC 7230, IETF RFC 7231, IETF RFC 7235, IETF RFC 5246, IETF RFC 8446). KMEs shall authenticate each request and identify the unique SAE ID of the calling SAE. Data in the message body of HTTPS requests from SAE to KME and HTTPS responses from KME to SAE shall be encoded in JSON format as per IETF RFC 8259.

This key delivery API is a REST-based API, a simple request and response style API between a SAE and a KME. Figure 6 shows how the key delivery API can be used for Multiple SAEs connected to a single KME. KME A and KME B exchange and store keys; each key delivered is assigned a universally unique ID. The test requirements for QKD Key delivery interface are defined in Chapter-2.

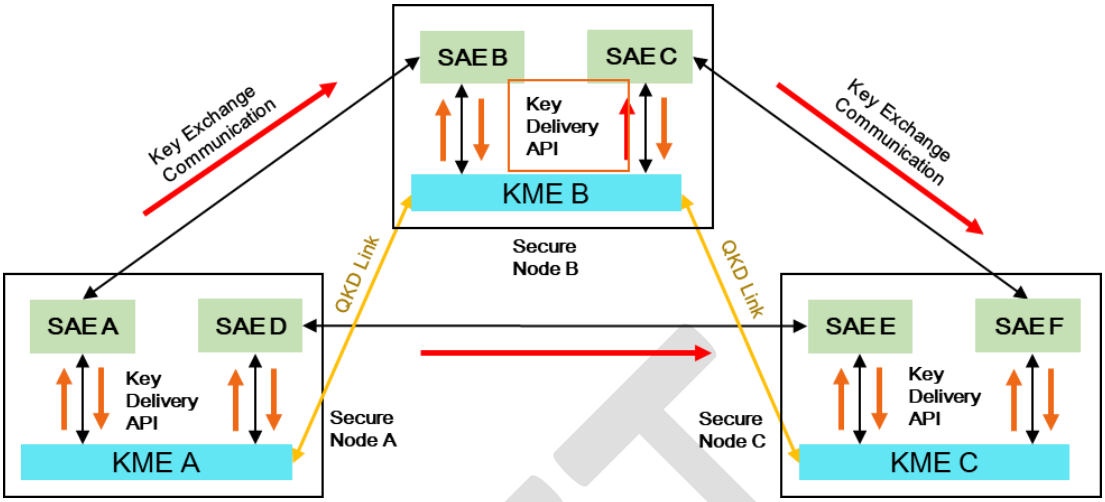

Figure 6: Block Diagram of communication flow of Key delivery management

2.1.10. Public and Private key pairs

A key pair is a set of two keys that work together as a team. In a typical key pair, you have one public and one private key.

List of functional test for key pair verification

Sl.No.	Test Category	Test Case	Expected Outcome
1.	Key Generation	Generate RSA 2048 key pair for classical system	Public and private keys are correctly formed and valid sizes.
		Generate ECC P-256 key pair for classical system	Keys match expected curve parameters and sizes.
		Generate ML-KEM (ML-KEM) key pair for Post Quantum System	Key pair generated with expected byte length (e.g. ML-KEM1024 public key = 1568 bytes).
		Generate ML-DSA (Dilithium) key pair for Post Quantum System	Public and private keys conform to NIST-specified sizes (e.g. ML-DSA3 public key = 1952 bytes).
2.	Key Validation	Verify key pair consistency for	RSA: Encryption with public key and decryption with

		classical system	private key must yield original message.
		Verify KEM decapsulation (PQC)	ML-KEM: shared secret from decapsulation must match the encapsulated one.
		Verify signature/verification pair (classical system)	Sign with private key, verify with public key; result must be valid for unchanged message.
		Verify signature/verification pair (for Quantum-safe)	Sign with ML-DSA (ML-DSA), verify using public key; ensure success.
3.	Negative test	Verify with mismatched public /private keys	Signature verification or KEM decapsulation must fail when using wrong key pair.
4.	Interop Test	Use key pair with external implementation (liboqs /Bouncy Castle)	Valid signature or shared key output across different libraries.
5.	Test Vectors	Validate against NIST test vectors (ML-KEM, ML-DSA) for Post Quantum System	Output of crypto operations (key gen, encapsulation, signing) must match known vectors.
6.	Performance	Measure keygen + sign/verify time	Timing within expected bounds for algorithm profile (e.g. ML-KEM keygen < 2ms).
7.	Hybrid Validation	Hybrid KEM: X25519 + ML-KEM shared key	Combined key derived successfully via concatenation or HKDF.

2.1.11. Post-quantum or Quantum-safe Algorithms

i. **Code-based cryptosystems**: The notion of code-based cryptography was first introduced by an encryption scheme published by McEliece in 1978. The McEliece cryptosystem builds on (binary) Goppa codes and their security based on the syndrome decoding problem. It is known to be extremely fast in encryption and reasonably quick in decryption. In the post-quantum context, HQC (Hamming Quasi-Cyclic) is another prominent code-based KEM candidate; it is based on quasi-cyclic codes and also relies on the hardness of decoding random linear codes, offering an alternative design point to Classic McEliece (typically trading much smaller keys for different parameter/performance characteristics).

List of functional requirements for code based systems

Sl. No.	Test category	Test cases	Expected outcome
1.	Protocol Conformance	Verify protocol messages follow specification format	All protocol messages comply with the protocol spec
2.		Validate handshake sequence correctness	Handshake completes without error; keys agreed on successfully
3.	Functional Correctness	Confirm key agreement results in matching shared secret	Both parties derive identical session keys
4.		Verify signature verification works on signed handshake	Signature validation succeeds for valid messages
5.	Error Handling	Test invalid or malformed messages	Protocol rejects malformed inputs gracefully without

			crashes
6.	Simulate replay attack detection	Protocol detects and rejects replayed messages	Simulate replay attack detection

ii. **Lattice-based cryptosystems**: Shortest Vector Problem (SVP) is to find the shortest non-zero vector within the lattice. SVP is known to be an NP-hard problem. The running time of solving a specific SVP instance remains to be discovered, i.e., it is still hard to estimate the exact computation of attacking a lattice-based cryptosystem. The security of the schemes is based on a lattice problem which is NP-hard under randomised reduction. And unlike the factorisation problem nor the discrete log problem, there is no known quantum-safe algorithm to solve SVP with the help of a quantum computer. Among all the candidates, the two algorithms are Learning With Error (LWE) based algorithms i.e. ML-KEM and ML-DSA. LWE is a mathematical problem widely used in lattice-based cryptography to create secure encryption algorithms to deliver the best performance and security. In practice, the Ring Learning With Error (R-LWE) variant is usually used to boost the efficiency of LWE-based systems. The security of the R-LWE problem reduces to the same lattice problem as SVP.

List of functional requirements for lattice based systems

Sl. No.	Test Description	Test Category	Input/output	Expected Outcome
1.	Key Generation correctness for LWE/R-LWE based scheme	Functional	Parameters for LWE/R-LWE (e.g., dimension, modulus)	Public/private keys generated correctly; satisfy R-LWE problem constraints
2.	Encryption and decryption correctness in LWE/R-LWE	Functional	Public key + plaintext	Decrypted message matches original plaintext; no errors during

	schemes			process
3.	Security against lattice attacks based on SVP hardness	Security	Multiple ciphertexts under same public key	Ciphertexts indistinguishable from random noise; no plaintext leakage
4.	Performance testing of LWE and R-LWE cryptosystems	Performance	Standard parameter sets and target hardware	Key gen, encrypt, decrypt within defined time limits (performance benchmarks)
5.	Resistance to fault injection and ciphertext corruption	Robustness/ Security	Inject faults into ciphertext or keys	System fails securely without revealing secret key or crashing
6.	Randomness and uniqueness of key generation	Consistency	Multiple key generations with same parameters	Generated keys are always valid and unique (no duplicates), satisfying randomness requirement
7.	Verify correct implementation of R-LWE efficiency boost	Functional/Performance	Implementations using ring structures (R-LWE variant)	Achieves efficiency gains over standard LWE while maintaining security

iii. Lattice-based signature Scheme: Lattice-based algorithms are faster and are considered quantum-safe. The security of lattice-based signature schemes relies on the presumed hardness of underlying lattice problems such as

(module) Learning With Errors ((M)LWE) and related (module) Short Integer Solution ((M)SIS)-type problems. These schemes are used to provide authentication, integrity, and non-repudiation through digital signatures, and are commonly deployed via standard certificate formats (e.g., X.509) and cryptographic libraries/providers.

List of functional requirements for lattice based signature systems

Sl. No.	Test Category	Test Description	Expected outcome
1.	Functionality	Verify key exchange using a configured post-quantum KEM (or hybrid KEM) during TLS handshake	Successful shared secret key establishment between client and server
2.	Signature Verification	Validate signature generation and verification with ML-DSA	Signature successfully generated and verified with valid keys
3.	Security	Test resistance against known quantum attacks	Cryptosystem remains secure under simulated quantum attack models
4.	Performance	Measure time taken for key exchange and signature generation	Operations complete within acceptable time limits (performance benchmark)
5.	Interoperability	Verify TLS connection establishment with PQ/hybrid key establishment and PQ signatures across	TLS connection established successfully without error

		supported TLS stacks/implementations	
6.	Fault Tolerance	Introducing corrupted signature and verify rejection behaviour	Invalid signatures are rejected, preventing authentication
7.	Mutual Authentication	Validate certificate-based mutual authentication with ML-DSA	Mutual authentication succeeds only with valid certificates

2.1.12. Hash based cryptosystems

Hash-based cryptography offers a one-time signature based on hash functions such as Lamport-Diffie or Winternitz signatures. The security of such one-time signature schemes relies solely on the collision resistance of the chosen cryptographic hash function.

List of functional requirements for hash based crypto systems

Sl. No.	Test Category	Description	Expected
1.	Functionality	Generate one-time signature using Lamport-Diffie and verify it	Signature successfully verifies with original message and public key
2.	Key Usage Limitation	Attempt to sign multiple messages with the same OTS key	Second signature fails or is flagged as insecure
3.	Security	Hash function collision resistance test	System remains secure as long as no hash collision occurs
4.	Signature Verification	Use altered message with valid signature	Signature fails to verify

5.	Performance	Benchmark signature generation and verification time	Operation completes within acceptable time limits
6.	State Management	Use stateful signature (e.g., XMSS) and reuse state incorrectly	System rejects or flags duplicate state usage
7.	Interoperability	Integrate hash-based signature (e.g., Winternitz) into TLS handshake	Signature integrates successfully in handshake if compliant with TLS constraints
8.	Robustness	Feed malformed public key into verification routine	Signature verification fails gracefully (not crash or verify incorrectly)

2.2. Crypto Agility

Crypto agility is the capability of a system, organisation, or infrastructure to rapidly and securely adapt its cryptographic mechanisms—including algorithms, parameters, keys, certificates, and protocols—without requiring major architectural redesign or service disruption. It enables the seamless introduction of new cryptographic algorithms, the coexistence of classical and post-quantum mechanisms (including hybrid constructions), and the timely retirement of deprecated or vulnerable algorithms in response to evolving threats, standards, or regulatory requirements. Crypto agility is a foundational requirement for post-quantum readiness, ensuring that cryptographic transitions can be managed as a controlled, repeatable lifecycle process rather than as disruptive one-time migrations.

2.2.1. Cryptographic Agility Requirements

The below cryptographic agility capabilities can be vendor and product-dependent. For example, a product may support cryptographic switching/configuration but may not support cryptographic negotiation.

- The system shall support cryptographic agility, enabling the addition, replacement, or deprecation of cryptographic algorithms with minimal architectural redesign.
- The system should support algorithm negotiation mechanisms covering classical, hybrid, and quantum-safe cryptographic algorithms.
- Cryptographic policies should be externally configurable (e.g., config file, API, or centralized policy server) and shall be protected against unauthorized modification (integrity protection + access control + audit logging)..
- The system shall support hybrid key establishment mechanisms that combine approved classical cryptographic algorithms with post-quantum cryptographic mechanisms.
- Hybrid key derivation mechanisms shall ensure cryptographic independence between classical and post-quantum components, such that compromise of one does not affect the security of the other. Forward secrecy should be preserved during hybrid operation, ensuring protection of past session keys even if long-term keys are compromised.
- Secure transport mechanisms should use TLS version 1.3 or higher with support for post-quantum cryptography-capable extensions. Legacy and insecure protocols, including TLS 1.0 and TLS 1.1, shall be disabled by default.
- Protocol downgrade attacks shall be prevented through appropriate cryptographic and protocol-level safeguards.
- The system shall support hybrid certificates that enable the use of both classical and post-quantum public keys and digital signatures. Certificate path validation shall function correctly for certificates employing hybrid and post-quantum cryptographic signatures.
- The system should support post-quantum cryptographic mechanisms in COSE-based environments, particularly for IoT, embedded systems, and deep packet inspection (DPI) use cases.
- The performance impact and resource utilization implications of post-quantum cryptographic mechanisms should be documented and optimized.

2.3. Quantum Safe Products based on different IETF Protocols

Since the IETF is still finalizing how PQC and PQC+traditional hybrid algorithms are integrated into widely used protocols (e.g., IPsec/IKEv2, TLS, SSH, and X.509/PKIX), therefore, conformance testing for such products can be performed manually using custom packet analyzers/decoders against the latest stable draft revisions, validating negotiation, algorithm identifiers, and handshake/message correctness as defined in those drafts which can later be done on published RFCs. Further, the hybrid schemes may also be tested as per RFCs (whether draft or published).

2.3.1. Hybrid X.509 certificates

X.509 defines public key certificates used to authenticate entities via signatures from publicly trusted authorities. These certificates are used in IETF's Public Key Infrastructure (PKI) X.509 (PKIX) standards and are widely deployed online for authentication. This describes a method of embedding alternative sets of cryptographic materials into X.509v3 digital certificates, X.509v2 Certificate Revocation Lists (CRLs), and PKCS #10 Certificate Signing Requests (CSRs). The embedded alternative cryptographic materials allow a Public Key Infrastructure (PKI) to use multiple cryptographic algorithms in a single object and transition to the new cryptographic algorithms while maintaining backward compatibility with systems using the existing algorithms. To use quantum-safe signatures with X.509, systems must support the new signature algorithms and their OIDs, and update certificate issuance and validation accordingly. These certificates can also authenticate the classical/service channel in QKD, ensuring QKD protocol messages remain authentic and tamper-evident.

S. No.	Test Category	Description
1.	Certificate Parsing & Format	Validate ASN.1/X.509 structure and hybrid extensions (e.g., multi-sig fields)
2.	Algorithm Binding Validation	Ensure correct linkage between classical and PQC signature algorithms
3.	Signature	Verify that both classical and PQC signatures

	Verification	can be independently validated
4.	Chain of Trust Testing	Test hybrid certificate path validation including mixed algorithm chains
5.	Interoperability Testing	Ensure certificates are compatible across browsers, servers, devices
6.	Fallback Handling	Tests on how systems handle PQC or classical verification failure
7.	Certificate Size Constraints	Evaluate bandwidth/storage impact due to large PQC signatures/keys
8.	TLS Handshake Integration	Ensure correct integration of hybrid certs in TLS 1.3 handshake
9.	Security Analysis	Check for downgrade or truncation attacks in hybrid settings
10.	Compliance & Policy Testing	Ensure certs align with CA/B Forum, NIST, and enterprise PKI policies

2.3.2. Internet Key Exchange version 2 (IKEv2)

Internet Key Exchange (IKEv2) is a protocol used to establish keys and Security Associations (SAs) to set up a secure Virtual Private Network (VPN) connection that protects network packets from being read or intercepted over a public Internet connection. The IKE protocol standard is rigid and does not permit VPN designers to choose beyond a small set of cryptographic algorithms. At present, the allowed algorithms are only partially quantum-safe. IKE provides authenticated connections using RSA, DSS or MAC with a pre-shared secret. IKE security associations are built on Perfect Forward Secrecy (PFS); in conventional security terms, ephemeral, one-time-use keys are created for every new secure connection. This ensures that the compromise of a long-term key only affects the confidentially of sessions established before the compromise. A replacement algorithm for the first and third exchanges, for instance, a quantum-safe alternative to replace the Diffie-Hellman key agreement to establish the shared secret for an IKE SA with perfect forward security. Together with a quantum-resistant authentication algorithm, this would enable IKE to

negotiate quantum-safe symmetric keys. QKDs or any quantum sourced/TRNG shared secrets may be used with conventional encryption ciphers or for one-time pad encryption in high-security applications. QKD or any quantum sourced/TRNG may also be used for the second pass to solve the key management problem of distributing shared secret keys for message authentication.

S. No.	Type test	Purpose
1.	Interoperability test	Ensure compatibility between different IKEv2 implementations using quantum-safe algorithms.
2.	RFC 8784 Conformance Testing	Validate support for mixing preshared keys with traditional keys for Post-quantum Security (hybrid key exchange).
3.	Performance Testing	Measure latency, CPU usage, and throughput when using post-quantum key encapsulation mechanisms.
4.	Cryptographic Algorithm Validation	Verify correct implementation of post-quantum algorithms (e.g., ML-KEM, ML-DSA).
5.	Resilience & Robustness Testing	Simulate attack scenarios (e.g., key compromise) to test fallback and recovery mechanisms.
6.	Key Exchange Integrity Verification	Confirm that negotiated session keys are consistent and derived securely using PQC algorithms.
7.	Protocol Downgrade Testing	Test defenses against downgrade attacks to pre-quantum algorithms.
8.	Long-Term Key Security Auditing	Audit long-term key storage and usage against quantum attack scenarios.

2.3.3. Transport Layer Security (TLS)

TLS is used to secure a variety of applications, including web traffic (the HTTP protocol), file transfer (FTP application) and mail transport (SMTP application). The design of TLS is mainly independent of cryptographic algorithms and allows parties to negotiate cipher suites (combinations of cryptographic algorithms to use). As of TLSv1.3, all cryptographic components (public key authentication, key exchange, hash functions, bulk encryption) can be negotiated, although generally, all must be arranged simultaneously in a single cipher suite rather than independently. Currently, most servers are authenticated using X.509 certificates containing RSA public keys and thus cannot be considered quantum safe.

A quantum-safe key exchange mechanism with perfect forward secrecy replaces existing key exchange mechanisms. To ease adoption, non-quantum-safe digital signatures, such as RSA, can continue to provide authentication. Quantum-safe cipher suites should match the security estimates of their symmetric primitives to the security estimates of their public key primitives. For example, a cipher suite utilising a quantum-safe public key algorithm at the 128-bit security level should use symmetric primitives at the 256-bit level to account for the impact of quantum search attacks.

Quantum-safe digital signatures can be deployed in certificates to authenticate the purely quantum-safe key exchange mechanism introduced in stage 1 above. A suitable mechanism for incorporating key material established from a quantum key distribution channel into TLS would allow parties to achieve high computational security from a relatively short QKD key.

S.No.	Test type	Purpose
1.	Hybrid Key Exchange Testing	Evaluate the integration of classical and post-quantum key exchange mechanisms in TLS 1.3.
2.	Performance Benchmarking	Assess the impact of PQC algorithms on TLS handshake latency, throughput, and resource utilization.
3.	Interoperability testing	Ensure compatibility between

		different TLS implementations using PQC algorithms.
4.	Embedded Systems Evaluation	Analyze the feasibility of PQC-enabled TLS on resource-constrained devices.
5.	Cryptographic Algorithm Validation	Verify the correctness and security of implemented PQC algorithms within TLS.
6.	Protocol Downgrade Resistance	Test the system's ability to prevent fallback to insecure, non-PQC algorithms.
7.	Certificate Verification Testing	Evaluate the handling of PQC certificates and signatures during the TLS handshake.

2.3.4. Secure/Multipurpose Internet Mail Extention (S/MIME)

It is a standard for digital signatures and public key encryption to send email messages securely. It offers origin authentication, non-repudiation, data integrity, and confidentiality through digital signatures and message encryption. This standard is widely adopted throughout government and enterprise. S/MIME, and a similar scheme called OpenPGP, allow email to remain encrypted during the entire path from the sender to the receiver. The most potent alternative to S/MIME for preserving end-to-end security is OpenPGP. Content encryption in S/MIME relies upon symmetric ciphers like AES that are believed to be quantum-safe. The above mentioned key establishment algorithms for these symmetric keys and the algorithms used for digital signatures are insecure in a Quantum-safe environment.

S. No	Test type	Purpose
1.	Hybrid Key Exchange Testing	Evaluate the integration of classical and post-quantum key exchange mechanisms in

		S/MIME.
2.	Performance Benchmarking	Assess the impact of PQC algorithms on S/MIME encryption and decryption speeds.
3.	Interoperability Testing	Ensure compatibility between different S/MIME implementations using PQC algorithms.
4.	Certificate Validation Testing	Verify the handling of PQC certificates and signatures during the S/MIME process.
5.	Compliance Checking	Ensure adherence to updated S/MIME Baseline Requirements, including support for PQC algorithms.

2.3.5. Secure Shell (SSH)

It is a secure remote-login protocol. It has pervasive and diverse applications and can be used for various purposes, including constructing cost-effective secure Wide Local Area Networks (WLAN), secure connectivity for cloud-based services, and essentially any other enterprise process requiring secure server access from a remote client. The SSH protocol involves three major sub-protocols: Transport Layer Protocol, the User Authentication Protocol, and the Connection Protocol. Each uses its algorithms to perform specific functions at different network layers. Within this protocol, several parameters are negotiated between server and client, including symmetric encryption algorithms, message authentication algorithms, and hash algorithms – all of which are quantum-safe. However, much like S/MIME, Key exchange and public key authentication methods rely upon insecure algorithms in the presence of quantum advantage. The following recommendations are suggested at the level of the Transport Layer Protocol:

2.3.5.1 Use of the Diffie-Hellman (DH) key exchange must be replaced by a quantum-safe algorithm that offers fast key-pair generation and perfect forward secrecy.

2.3.5.2 The use of the Digital Signature Algorithm (DSA), the Elliptic Curve Digital Signature Algorithm (ECDSA) and the RSA Signature Scheme

Algorithm (RSA-SSA) for host authentication must be replaced by the use of quantum-safe authentication mechanisms such as quantum-safe digital signatures or message authentication codes based on a pre-shared symmetric key.

2.3.5.3 Quantum Key Distribution is one of the viable methods for secret key generation within the SSH protocol. Using QKD would bypass issues related to the presently unsafe practices of private key exchange and could replace the current key-establishment methods for symmetric (AES) keys.

S.No.	Test type	Purpose
1.	Hybrid Key Exchange Testing	Evaluate the integration of classical and post-quantum key exchange mechanisms in SSH.
2.	Performance Benchmarking	Assess the impact of PQC algorithms on SSH handshake latency, throughput, and resource utilization.
3.	Interoperability Testing	Ensure compatibility between different SSH implementations using PQC algorithms.
4.	Cryptographic Algorithm Validation	Verify the correctness and security of implemented PQC algorithms within SSH.
5.	Protocol Downgrade Resistance	Test the system's ability to prevent fallback to insecure, non-PQC algorithms.
6.	Configuration Compliance Checking	Ensure SSH configurations align with quantum-safe standards and best practices.

2.4. Endpoint devices

Endpoint devices include any piece of hardware that a user utilises to interact with a distributed computing system or network. These can include canonical

examples such as personal computers and mobile phones, kiosks/terminals in banks, stores, and airports, and any embedded technology connected to a broader network. Encryption and authentication of endpoint devices refer to making the contents of the device unreadable to unauthorised parties through cryptography and security protocols. This mechanism is a critical practice to prevent unauthorised data transfer and access, to ensure that only approved devices are allowed access to the system, and to deal appropriately with rogue or compromised devices that threaten system security through intrusions such as malware, key loggers, or viruses.

Test Case ID	Test Category	Description	Expected Outcome
1.	Secure Key Exchange	Use encrypted key exchange during endpoint handshake	Secure session key is generated and transmitted securely to authorized device
2.	Device Data Protection	Encrypt local storage on endpoint with lattice-based hybrid encryption	Data is unreadable to unauthorized parties; decryption only with valid key
3.	Session Resilience	Interrupt session during exchange	Session either securely resumes or fails gracefully without data leakage
4.	Performance (Device)	Measure CPU/memory usage during lattice-based handshake	Handshake completes within acceptable resource limits on standard devices

2.5. IOT based Products (Lightweight Cryptography)

Storage servers and data must be secure throughout their entire transfer

through a network from one location. The security of resource-constrained devices is critical in the IoT field, given that everything is interconnected. The concern is that the limited resources on these devices may cause performance issues when the standard cryptographic algorithms are running on them. Therefore, in recent years, researchers have been working on developing lightweight cryptography and various efficient cryptographic technologies. Its requirements are constrained by security, low cost and high performance.

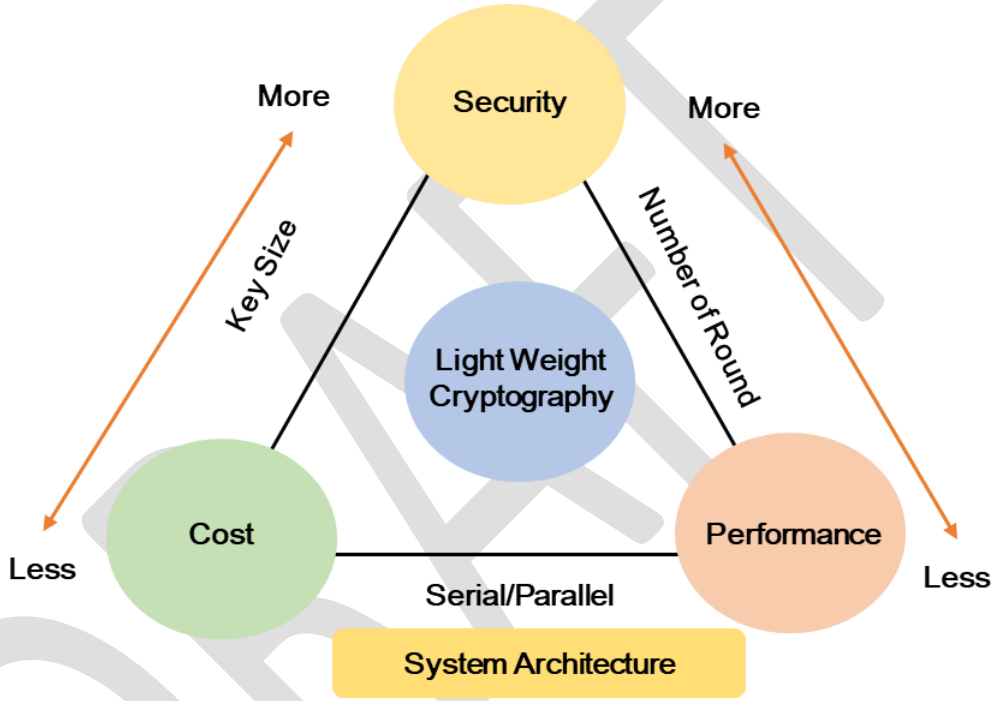

Figure 7: Block Diagram of Lightweight cryptography design trade-offs.

These requirements are balanced accordingly by adjusting the key size, the number of encryption rounds and the system architecture. Thus, the target of lightweight cryptography is to find a better balance between performance and security within cost constraints (refer Figure 7). The chosen algorithms are designed to protect information created and transmitted by the Internet of Things (IoT), including its myriad of tiny sensors and actuators. They are also designed for other miniature technologies, such as implanted medical devices, stress detectors inside roads and bridges, and keyless entry fobs for vehicles.

Devices like these need "lightweight cryptography" protection that uses the limited amount of electronic resources they possess.

The most important in lightweight cryptography: authenticated encryption with associated data (AEAD) and hashing.

AEAD protects the confidentiality of a message, but it also allows extra information, such as the header of a message, or a device's IP address, to be included without being encrypted. The algorithm ensures that all of the protected data is authentic and has not changed in transit. AEAD can be used in vehicle-to-vehicle communications, and it also can help prevent the counterfeiting of messages exchanged with the Radio Frequency Identification (RFID) tags that often help track packages in warehouses. They need to compliant NIST protocols as listed from time to time as per the user requirements.

Sl. No.	Test Category	Description	Expected outcome
1.	AEAD Functionality	Verify ~~lattice-based~~ AEAD encryption and decryption with associated data (e.g., header)	Data and associated data are encrypted and authenticated correctly
2.	Confidentiality	Ensure message confidentiality over noisy IoT channel using ~~lattice~~ AEAD	Encrypted data not recoverable by unauthorized parties
3.	Authenticity	Test integrity verification when associated data is tampered with	Verification fails, rejecting tampered messages
4.	Key Size Trade-off	Evaluate security vs performance by adjusting ~~lattice~~ key sizes	Smaller keys increase speed but maintain acceptable security levels

5.	Resistance to Attacks	Simulate side-channel and fault injection attacks on ~~lattice~~ AEAD	System withstands attacks without key leakage or authentication bypass
6.	Performance	Measure encryption/decryption speed and power consumption on IoT hardware	Cryptosystem completes operations within device constraints
7.	Compliance Testing	Confirm compliance with NIST Lightweight Cryptography guidelines	Algorithm passes all mandatory compliance tests

In addition to AEAD and hashing, constrained IoT deployments often require digital signatures for firmware authenticity, secure boot / measured boot attestations, device identity, and non-repudiation in audit logs. FN-DSA or FIPS-206 is a digital signature algorithm that can be used for digital signature requirement for constraint devices.

Sl. No.	Test Category	Description	Expected outcome
1	Signature Functionality	Generate signature and verify signature over representative IoT messages (telemetry, control commands)	Valid signatures verify; invalid signatures fail
2	Message/Context Binding	Verify signatures bind to correct context (device ID, protocol header, domain separation tag if used)	Replay across contexts fails; correct context verifies
3	Negative Testing	Verify behavior for modified message, modified signature,	Verification fails reliably without crashes

		wrong public key, truncated inputs	
4	Key Generation	Test key-pair generation across supported parameter sets; verify key validity checks	Keys generated correctly; invalid keys rejected
5	Robustness (Malformed Inputs)	Feed malformed/edge-case encodings (oversized, invalid length, non-canonical forms)	Implementation rejects safely; no memory errors
6	Side-channel Resistance	Evaluate timing/power leakage under typical signing and verify operations (where applicable)	No exploitable leakage beyond acceptable threshold
7	Fault Injection Resilience	Simulate fault conditions during signing/verification (glitches, bit flips)	No key leakage; faulty signatures do not verify
8	Performance on IoT HW	Measure sign/verify time, RAM/Flash footprint, and energy consumption	Operations within device constraints; meets procurement limits
9	Interoperability	Interop test with a known-good reference implementation across parameter sets	Cross-implementation verification succeeds
10	Compliance Testing	Confirm compliance with applicable NIST/IETF profiles and test vectors	Passes mandatory vectors and profile requirements

Further, ASCON is a lightweight cryptographic family designed for constrained devices and selected by NIST for lightweight cryptography. For IoT, ASCON AEAD provides confidentiality + integrity with associated data (headers, addresses, counters), while ASCON-Hash supports integrity checks,

commitments, and protocol hashing needs.

Sl. No.	Test Category	Description	Expected outcome
1.	Algorithm ID / Variant	Verify the implementation supports the intended standardized functions: Ascon-AEAD128, Ascon-Hash256, Ascon-XOF128, Ascon-CXOF128 (as applicable).	Implemented variants match the claimed functions; unsupported variants are not claimed.
2.	AEAD Correctness	Verify ASCON AEAD encrypt/decrypt with associated data using official test vectors	Ciphertext and tags match vectors; decrypt recovers plaintext
3.	Associated Data Integrity	Modify AD (header/IP/metadata) and verify tag failure	Verification fails; plaintext not released
4.	Nonce Handling	Verify unique-nonce requirement enforcement (generation/storage/anti-reuse)	No nonce reuse in normal operation; reuse detected/mitigated per policy
5.	Tag Verification Behavior	Ensure constant-time tag verification and "fail closed" behavior	No timing oracle; invalid tags always rejected
6.	Misuse / Edge Cases	Test zero-length plaintext/AD, maximum lengths, fragmented inputs	Correct outputs; no crashes; consistent streaming behavior if supported
7.	Replay Protection Support	Validate integration with counters/timestamps as AD (where protocol uses	Replayed messages detected by system logic; crypto validates

		it)	binding
8.	Hash Correctness	Verify ASCON-Hash outputs for standard vectors and typical IoT payloads	Hash outputs match vectors; stable across platforms
9.	XOF Functional Correctness	Validate Ascon-XOF128 supports selectable output length L > 0, correct number of squeezed blocks and correct IV usage.	Output length equals requested L; output matches known-good implementation for various L.
10.	CXOF Customization Handling	Validate Ascon-CXOF128: customization string input Z is supported and incorporated as specified; verify the length constraint	--
11.	Robustness (Malformed Inputs)	Corrupt lengths/encodings and provide malformed buffers	Implementation rejects safely; no memory corruption
12.	Side-channel Resistance	Timing/power analysis on encryption/decryption and hashing (if required)	Leakage within acceptable limits; masking/constant-time validated
13.	Performance on IoT HW	Measure throughput, latency, RAM/Flash, and power on target MCU/SoC	Meets device constraints and procurement targets
14.	Compliance Testing	Confirm conformance to NIST LWC ASCON specifications and published vectors	Passes all mandatory compliance tests

2.6. Cloud based PQC products (Cryptography –as – a- service)

Cloud services have become ubiquitous due to the rise of high-capacity networks, the decreased cost of computers and data storage devices, and trends toward hardware virtualisation and infrastructure, platform, and software-as-a-service models..

2.6.1. Cryptography-as-a-Service

Deploying cryptographic keys to endpoints such as smartphones, virtual machines in the public cloud and smart grid equipment is risky. Therefore, this proposes a Cryptography as a Service (CaaS) model, which allows cryptographic operations to be performed without exposing cryptographic keys and recommends overcoming the pitfalls associated with this technology. Keyed cryptographic operations, such as encryption and decryption, are performed by a CaaS provider on behalf of a device via web services APIs. Cryptography as a service has been defined as being "Keyed cryptographic operations, such as encryption and decryption that are performed by a CaaS provider on behalf of a device via web service APIs". The way that the "as a service" architecture works is through the implementation of HTTP and systems such as REST and SOAP. The overall architecture is extremely similar to Public Key Authorities (PKA) and Certificate Authorities (CA). The cryptographic keys used to perform these operations are stored within the CaaS provider, so devices do not possess these keys at any time (refer figure 8).

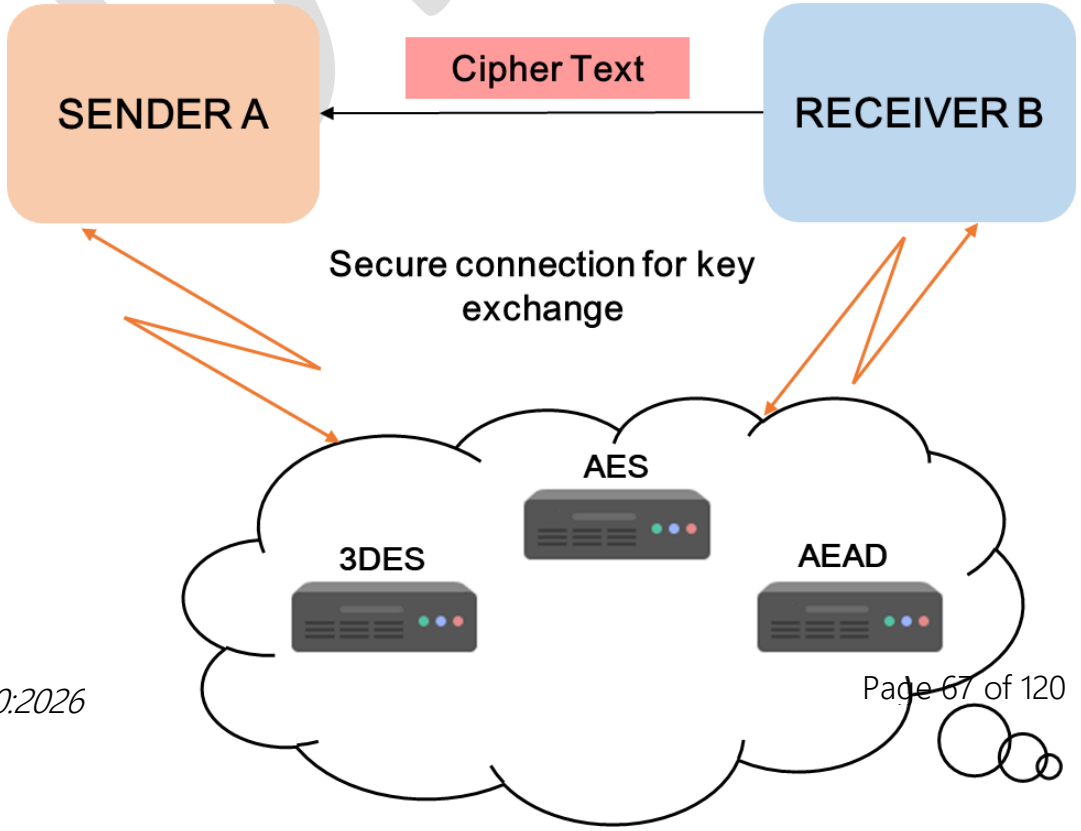

Figure 8: Block Diagram - Cryptography-as-a-Service

2.6.2. Cryptography Service Provider (CSP)

As organisations continue to test and integrate cloud computing into their IT environments, Cryptography-as-a-service and Entropy-as-a-service are into service to safeguard cryptographic keys with the same dynamic and virtualised attributes of cloud computing environments. Additionally, when storing data in multi-clouds, using native encryption from cloud service providers creates silos of data and the risk of not having full control over your keys and data. On-premises Hardware-Secure-Module(HSM) diminish those silos and enable users to know the whereabouts of their keys at all times.

BYOE (bring your encryption), or BYOK (bring your keys), is a security model tailored explicitly to cloud computing. It allows cloud service customers to use their encryption tools and manage their encryption keys. A cryptographic Service Provider (CSP) allows Cryptographic applications and services to access secure cryptographic operations and Key management. This provider uses the standard REST API, JCE (Java Cryptographic Extension) programming interface. PKCS#11, Cryptography API: Next Generation (CNG), HTTPS, Web API (W3C), Microsoft CAPI, and OpenSSL.

2.6.3. Verification of cloud/service based key lifecycle management

2.6.3.1 Verify integration of Cloud HSM (e.g., AWS CloudHSM, Azure Key Vault, Google Cloud KMS etc.)

2.6.3.2 All Cloud HSM deployments shall support complete cryptographic key lifecycle management, including secure key generation, storage, usage, rotation, archival, and destruction.

2.6.3.3 Cryptographic keys shall be generated and remain within FIPS 140-2 Level 3 (or higher) validated HSM security boundaries, and plaintext export of key material shall not be permitted from the validated HSM security boundaries.

2.6.3.4 Access to keys shall be governed by role-based access control, enforcing least-privilege, segregation of duties and multi-factor authentication.

2.6.3.5 Key rotation policies shall be mandatorily enforced, with keys rotation periods as

per NIST SP 800-57 Part 1 Rev. 5. The key rotation periods shall be shorter for high-risk systems. Longer crypto period is allowed for Root / Master keys but periodic rotation is recommended.

2.6.3.6 Automated rotation mechanisms shall be supported without service disruption, and previous key versions shall remain available for decryption or verification only.

2.6.3.7 All key lifecycle events—including creation, access, rotation, policy changes, and destruction—shall generate immutable audit logs.

2.6.3.8 Audit logs shall be tamper-evident, exportable to external systems with audit log retention periods as defined in alignment with NIST SP 800-57, NIST SP 800-53, ISO/IEC 11770, ISO/IEC 27001, and applicable national regulatory requirements, including CERT-In cyber security directions. For eg. – The Audit logs may be retained online for a minimum of 400 days, and archived securely for a minimum period of seven years.

2.6.3.9 Cryptographic destruction of keys shall be irreversible and verifiable through audit evidence.

2.6.3.10 Check secure key provisioning, distribution, rotation, archival, and destruction mechanisms as per above steps.

2.6.3.11 Verify that only authorized services or users can access HSM APIs and review IAM policies and role-based access configurations.

2.6.3.12 Perform dynamic tests on key management APIs for:

- Unauthorized access attempts
- Replay attacks and injection vulnerabilities
- Improper error handling revealing sensitive info
- Data-in-Transit and Data-at-Rest Protection

2.6.3.13 Verify Cloud HSM complies with FIPS/ISO standards (or equivalent) at least FIPS 140-2 Level 3 (or higher) validated HSMs.

2.7. Security Services

Encryption is vital in protecting sensitive data transmitted over an unsecured network or stored at rest in computer systems. During the transfer of data over

an unsecured network, a cryptographic system should ensure the following security services to ensure the security of the system or data transmission.

2.7.1. Approved Confidentiality Technique:

The data in network traffic must be available only to the intended recipient. In other words, the data in network traffic must not be available to anyone other than the intended recipient.

2.7.2. Approved Integrity Technique:

The data in network traffic must not be altered while in a network. In other words, the recipient's data must be the same as the data sent by the Sender.

2.7.3. Approved Authentication Technique:

The Sender and the Recipient must prove their identity to each other.

2.7.4. Access Control:

The principle of access control decides who should be capable of accessing information or a system through a communication link. It supports the avoidance of unauthorised use of a resource.

2.7.5. Non-repudiation:

Non-repudiation prevents either sender or receiver from adverse a transmitted message. Therefore, when a message is sent, the receiver can validate that the asserted sender sent the message. Similarly, when a message is received, the sender can validate that the asserted receiver received the message.

CHAPTER-3

Operational, Interface and Interoperability Requirements

3.1 Based on network deployment topologies, the cryptographic system should work in point-to-point/ point-to-multipoint / multipoint-to-multipoint mode.

3.2 The cryptographic system shall provide Ethernet payload encryption over a point-to-point network.

3.3 It must be possible for an operator to select a particular encryption scheme for payload encryption system wise.

i. It shall provide confidentiality and protection from firmware upgrades.

ii. It shall support Policy based encryption.

iii. It shall provide data protection against unauthorised access by users and processes in physical, virtual, and cloud environments so that implementation is seamless and transparent to application/presentation of layer of system and its storage. So it can work across an enterprise's entire environment.

iv. Regardless of performance level, the cryptographic system shall be interoperable with the appropriate Application interface.

v. It shall provide confidentiality using standard encryption algorithms in a Quantum-safe cryptosystem and applicable algorithms in asymmetric and hash functions as per the product's specification sheet.

~~vi. It may support encryption through a proprietary encryption algorithm also.~~

3.4 Operational requirements of a cryptographic system

Sl. No	Parameter Type	Description and range of the Parameters	Reference Standard(s)	Remarks
1	Traffic type	Unicast / Multicast / Broadcast over IP networks (IPv4 / IPv6)	IPv4: RFC 791, IPv6: RFC 8200	To be confirmed as per applicable RFCs and product scope
2	No of Concurrent connection	User-to-Server mode: support at least 100/500 connections as applicable	TCP: RFC 9293, TLS 1.3: RFC 8446	Exact value to be finalized by procurer based on deployment sizing; verification via load/concurrency testing
3	Direction of data transmission	Full duplex	IEEE 802.3 (Ethernet), as applicable	Low overhead bits
4	Separation of data/control plane	Separation of Control plane and data plane	Product architecture / deployment specification	Physical and logical separation of data and control plane
5	Latency at specified rate (server/client)	Latency at node (non-aggregation state) ≤ 10 µs for data throughput up to 10 Gb/s	--	Latency requirement to be independent of packet/Ethernet frame size
6	Support of Jumbo frames	Support Ethernet frames larger than standard MTU; configurable jumbo frame size	IEEE 802.3	Beyond standard ethernet frame size, exact maximum MTU to be specified by procurer
7	Mode of secure key uploading	Manual/Automatic	ISO/IEC 19790:2025	Applicable according to security level 1/2/3/4

8	Software/ Firmware loading	The cryptographic module can load software or firmware from an external source.	ISO/IEC 19790:2025 Clause 7.4.3.4	A validation authority shall validate the integrity/authenticity of loaded software or firmware before loading.
9	Self-test for the integrity of H/W and S/W modules	Cryptographic module pre-operational and conditional self-tests provide the operator assurance that faults have not been introduced that would prevent the module's correct operation.	ISO/IEC 19790:2025 Clause 7.10	Conditional self-tests shall be performed when an applicable security function or process is invoked.
10	Module's version	The cryptographic module shall output the name or module identifier and the versioning information	ISO/IEC 19790:2025 Clause 7.4.3.1	Hardware, software and/or firmware versioning information
11	Status	The cryptographic module shall output the current status	ISO/IEC 19790:2025 Clause 7.4.3.1	Visual indicators in response to a service request/ normal state
12	Self-tests	pre-operational self-tests before loaded code can be executed	ISO/IEC 19790:2025 Clause 7.4.3.1	Status shall reflect completion/pass/fail
13	Approved Security function test	Approved security functions	ISO/IEC 19790:2025 Clause 7.4.3.1	at least one test in the approved mode of operation
14	Zeroisation	Perform zeroisation (zeroise all unprotected SSPs and	ISO/IEC 19790:2025 Clause 7.4.3.1	Zeroisation shall be immediate and uninterruptable in

		key components within the module at all security levels)		Security Level 4
15	Mode of operation	Normal/degraded	ISO/IEC 19790:2025 Clause 7.2.4.3	Normal mode only if all pre-operational self-tests pass
16	Bypass capability	Indicate whether the Bypass capability is activated or not	ISO/IEC 19790:2025 Clause 7.2.4.3	Bypass allowed only with controls preventing inadvertent plaintext exposure due to a single error
17	Self-Initiated cryptographic output Test	Indicate capability for self-initiated crypto output test without Crypto Officer configuration; show status when enabled	ISO/IEC 19790:2025	this configuration may be preserved over resetting, rebooting, or power cycling of the module
18	Operational environment	i. A non-modifiable operational environment ii. A limited operational environment. A modifiable operational environment	ISO/IEC 19790:2025 Clause 7.6	Functions may be added or modified within the operational environment.
19	Life-cycle assurance	Confirm the best practices by the vendor of a cryptographic module during the design, development, operation and end of life of a cryptographic module.	ISO/IEC 19790:2025 Clause 7.11	Vendor to provide evidence covering defined lifecycle stages

20	Power	AC supply 110-230V +10% 50/60 Hz AC	Vendor specification	AC or DC supply or both as optional Procurer to specify connector type and redundancy if required
21	DC power	DC supply range: – 40 V to –60 V DC (including renewable sources, as applicable)	Vendor specification	
22	Size	Dimensions in mm or inches in length, width and height	Vendor specification	Desirable: 1U; multiple 1U options acceptable
23	Cooling	Requirement of Ingress or Egress fans (suck and exhaust kind of setup).	--	Fan may be optional depending on environment; temperature maintenance is mandatory
24	Min Altitude without any degradation	Normal operation without any degradation at an altitude of upto 3,000 meters.	--	The manufacturer shall guarantee satisfactory performance
25	Power Supply Alarm	Visual indicator (e.g., Green/Red) for AC/DC power status		Indicate the status of power AC/DC.
26	Encryption/Decryption Alarm	Any visual indicator(G/R or any other colour)		To indicate status
27	Fault Indicator Alarm	Any visual indicator(G/R)	Log message and visual indicator	To indicate status

28	Capable of functioning in a saline environment	Without any degradation system shall be able to function normally		Self-certificate to be submitted if no test environment is available.

3.5 Interface requirements of a cryptographic system

The cryptographic system shall support 10/100/1000/2500/10000 BASE-TX electrical or optical interface or any open standard port for management as per the user requirement. Hardware/Software of Plaintext Interface shall be physically separate from Hardware/Software of Cipher interface.

Sl. No	Name of the Sub parameter	Types of Parameters range	Reference Standard(s)	Remarks
1	Management Interface	Optical//Ethernet (RJ45) Ethernet data input through the command line interface also. SNMP v3 or above, or XML/JSON shall be supported for EMS/NMS/NOC.	ISO/IEC 19790:2025 Clause 7.3.1, 7.3.2	Applicable interface type(s) depend on module category: (i) HMI (Hardware Module Interface): data port + management port; (ii) SFM (Software/Firmware Module): logical interfaces; (iii) HSMI/HFMI (Hybrid SW/FW Module Interface): plaintext/ciphertext interface separation
2	Data input interface	Interface (plain text, cipher text and SSP)	ISO/IEC 19790:2025 Clause 7.3.3(a)	
3	Data output interface	Interface (plain text, cipher text and SSP)	ISO/IEC 19790:2025 Clause 7.3.3(b)	

4	Control input interface	All input commands, signals, and control data (e.g., clock input, function calls, manual controls such as switches/ buttons/ keyboards)	ISO/IEC 19790:2025 Clause 7.3.3(c)	Access control/authentication applies as per module security policy
5	Control output interface	All output commands, signals, and control data	ISO/IEC 19790:2025 Clause 7.3.3(d)	Inhibited when the cryptographic module is in an error state unless exceptions are specified
6	Power interface	All external electrical power entering/leaving the cryptographic module (not applicable to pure software-only modules)	ISO/IEC 19790:2025 Clause 7.3.3(g)	Except in the software module, power is provided internally by the source of the battery.
7	Status output	All status output signals, indicators, and status data (including visual/audio/mechanical indicators)	ISO/IEC 19790:2025 Clause 7.3.3(e)	Includes error indicators (return codes), displays/LEDs, buzzer/tone/ring/vibration where implemented
8	Trusted channel (Security Level 3 and above)	Protected link for transmission of unprotected plaintext CSPs/key components and authentication data between module and endpoint	ISO/IEC 19790:2025 Clause 7.3.4	For Security Level 4, multi-factor identity-based authentication shall be employed for all services utilising the trusted channel

4.6. Interoperable requirements of a cryptographic system

Interoperability is one of the essentials to making seamless internetwork function in a heterogeneous network environment. The application service layer in the cryptographic system communicates with the key management controller. Communication protocol and data format for a quantum key distribution (QKD) network or any Key source network to supply cryptographic keys to an application, i.e., a Cryptographic system.

Sl. No	Name of the Sub parameter	Types of Parameters range	Reference Standard(s)	Remarks
1	IP Layer	Internet Protocol (IP) IPV4/IP6, Internet Control Message Protocol (ICMP), Internet Group Management Protocol (IGMP), IPSec	IETF RFCs IPv4: RFC 791; IPv6: RFC 8200; ICMPv4: RFC 792; ICMPv6: RFC 4443; IGMP: RFC 3376; IPsec: RFC 4301 (and related RFCs)	Confirm interworking for both IPv4 and IPv6;
2	Authentication	CA-based authentication and/or AAA-based authentication (as applicable)	X.509/PKI; RADIUS: RFC 2865; TLS: RFC 8446	CA trust model as per (enterprise CA / public CA / local RA); RADIUS server may be used for centralized AAA
3	Encryption	Various encryption methods as listed	NIST-approved algorithms (as applicable)	Device shall support configured algorithms/cipher suites as per security policy
4	Key exchange (KMIP)	During a key exchange with other systems	OASIS standard	--

5	API with QKD interface	REST/HTTPS API for middleware integration; JSON encoding	HTTPS/TLS: RFC 8446; JSON: RFC 8259	--
6	Inter Secure Application Entity (SAE)	Master SAE ↔ Slave SAE communication for QKD key delivery/control	QKD link as per ETSI GS QKD 004	SAE of cryptographic system connects to the QKD KME
7	SSH	User authentication layer, transport layer, connection layer	RFC 4252, RFC 4253, RFC 4254	SSH may be used in several methodologies like for secure administration/management access
8	TLS	TLS v1.3 or above	IETF RFC 8446	Use for management/API channels and other interfaces requiring secure transport
9	Entropy source	Proven/validated randomness source for key generation and nonce	NIST SP standards, TEC QRNG standard	Examples: on-chip TRNG, dedicated circuitry, external entropy source;
10	Clock	Internal circuit or External I/O timing source	Product/platform spec	Used for control functions, timeouts, scheduling, counters/ ticks; accuracy requirements to be specified if needed
11	Link layer protocols	L2 options such as Ethernet MAC, PPP, tunneling (as applicable)	IEEE 802.3; PPP: RFC 1661; relevant tunnel RFCs as applicable	Layer 2 protocol communications

CHAPTER-4

Security Requirements

4.1. Security services requirements of a cryptographic system

The following security services are required for the enhancement of security;

(i) Authentication mechanisms may be needed within a cryptographic module to authenticate an operator accessing the module and to verify that the operator is authorised to assume the requested role and perform services within that role. The cryptographic system shall support lossless data encryption/ decryption key change.

(ii) It should implement a key integrity check and authentication mechanism through a suitable hashing algorithm.

(iii) Encryption keys should be encrypted, stored in a secure device and only accessible to the user, regardless of data and key storage methods.

4.2. Security/Assurance level classification

The cryptographic techniques (algorithms and protocols) may remain the same across different security/assurance levels; however, the assurance requirements increase with the risk category, usage environment, and deployment criticality. Selection of a PQC-based cryptographic solution shall therefore be based on the assurance level appropriate to the application's risk appetite and the operational environment. The hierarchical structure defines that higher assurance implies compliance with lower assurance requirements, and the quantum-safe cryptographic systems are classified into Level 1, Level 2, Level 3, and Level 4 based on security needs and deployment context..

(i) Security/Assurance Level 1: Provides a baseline level of security for PQC adoption in non-sensitive, consumer-grade environments. Focus is on basic PQC compliance, correctness, interoperability (including RFC conformance where applicable), and basic performance checks. Basic security requirements are specified for a cryptographic module (e.g. at least one approved security function or approved sensitive security parameter establishment method shall be used). Ideally appropriate for security applications where controls, such as physical security, network security, and administrative procedures, are provided outside the module but within the deployable environment.

(ii) Security/Assurance Level 2: Applies to PQC products/solutions handling sensitive data in consumer-grade environments, including cloud-integrated implementations. In addition to Level 1 checks, it emphasizes secure software assurance such as robustness, fuzz/negative testing, vulnerability assessment, and secure coding practices. It enhances the physical security mechanisms of Security Level 1 by adding the requirement for tamper evidence, including tamper-evident coatings or seals or pick-resistant locks on removable covers or doors. Security Level 2 allows a cryptographic software module to be executed in an adaptable environment that implements role-based access controls or, at the minimum, a discretionary access control with the robust mechanism of defining new groups and assigning restrictive permissions through access control lists (e.g. ACLs), and with the capability of setting each user to more than one group, and that protects against unauthorised execution, modification, and reading of cryptographic software.

(iii) Security/Assurance Level 3: Applies to enterprise-grade deployments requiring long-term security for sectors such as finance, telecom, and health. In addition to Level 2 requirements, it introduces stronger enterprise controls such as crypto-agility validation, stronger entropy validation expectations (TRNG/QRNG as applicable), centralized cryptographic management integration, and broader enterprise security assurance practices. It provides additional requirements to mitigate unauthorised access to SSPs held within the cryptographic module. Physical security mechanisms required at Security Level 3 are intended to have a high probability of detecting and responding to attempts at direct physical access, use or modification of the cryptographic module and probing through ventilation holes or slits. The physical security mechanisms may include solid enclosures and tamper detection/response circuitry that zeroise all CSPs when the removable covers/doors of the cryptographic module are opened. Security Level 3 requires identity-based authentication mechanisms, enhancing the security provided by the role-based authentication mechanisms specified for Security Level 2. A cryptographic module authenticates the identity of an operator and verifies that the identified operator is authorised to assume a specific role and perform a corresponding set of services. Security Level 3 requires manually established plaintext CSPs to be encrypted, utilise a trusted channel or use a split knowledge procedure for entry or output.

(iv) Security/Assurance Level 4: Applies to sovereign-grade / critical information

infrastructure protection. This level represents the most stringent assurance, including the strongest security assurance expectations (e.g., strategic resilience, rigorous supply chain assurance, nation-state attack simulation–type security validation), and sector-specific requirements. The physical security mechanisms provide a complete envelope of protection around the cryptographic module to detect and respond to all unauthorised attempts at physical access when SSPs are contained in the module, whether external power is applied or not. Penetration of the cryptographic module enclosure from any direction is highly likely to be detected, resulting in the immediate zeroisation of all unprotected SSPs. Security Level 4 introduces the multi-factor authentication requirement for operator authentication. At a minimum, this requires two of the following three attributes. At Security Level 4, a cryptographic module is required to include special environmental protection features designed to detect voltage and temperature boundaries and zeroise all unprotected SSPs to provide a reasonable assurance that the module will not be affected when outside of the normal operating range in a manner that can compromise the security of the module.

Security services requirements of a cryptographic system

Sl. No.	Parameter	Security Level-1	Security Level-2	Security Level-3	Security Level-4	Reference standards
1	Cryptograhic Module Interfaces	Required and optional interfaces. Specification of all interfaces and all input and output data paths.		Trusted channel		ISO/IEC 19790:2025
2	Roles, Services, and Authenticatio n	Logical separation of required and optional roles and services.	Role-based or identity-based operator authenticati on.	Identity-based operator authenticatio n.	Multi-factor authenticati on.	ISO/IEC 19790:2025
		Approved integrity technique,	An approved			

3	Software/Firmware Security	or EDC-based integrity test. Defined SFMI, HFMI and HSMI.	digital signature or keyed message authentication code-based integrity test.	Approved digital signature-based integrity test.		ISO/IEC 19790:2025
		Executable code.				
4	Operational Environment	Non-Modifiable, Limited or Modifiable.	Modifiable.	Non-modifiable		
		Control of SSPs.	Role-based or discretionary access control. Audit mechanism			
			Tamper evidence.	Tamper detection and		

5	Physical Security	Production-grade components.	Opaque covering or enclosure.	response for covers and doors. Strong enclosure or coating. Protection from direct probing. EFP or EFT.	Tamper detection and response envelope. EFP. Fault injection mitigation.	ISO/IEC 19790:2025
6	Non-Invasive Security	The Module is designed to mitigate against non-invasive attacks. Documentation and effectiveness of mitigation techniques specified for security classes 1&2. Mitigation testing is essential in security classes 3&4.				ISO/IEC 19790:2025
7	Sensitive security parameter generation	Random bit generators, SSP generation, establishment, entry and output, storage, and zeroization. Automated SSP transport and SSP agreement using approved methods.				ISO/IEC 19790:2025
		Manually established SSPs may be entered or output in plain text		Manually established SSPs may be entered, or output in encrypted form via trusted channel or split knowledge procedures		
8	Self-Tests	Pre-operational: Software/firmware integrity, bypass and critical functional test.				ISO/IEC 19790:2025
		Conditional: Cryptographic algorithm, pair-wise consistency, Software/firmware loading, manual entry, bypass and critical functional test.				

9	Mitigation of other attacks	Specification of Mitigation of attacks for which no testable requirements are available currently			Specification of Mitigation of attacks with testable requirements	ISO/IEC 19790:2025
10	Replay attacks					To be verified
11	Fault injection Attacks					To be verified
12	timing-based side-channel attacks					To be verified
13	Man-in-the -middle Attack					To be verified
14	Documentation and validation	Applicable	Applicable	Applicable	Applicable	ISO/IEC 19790:2025
15	Cryptographic Checks	Verify PQC (e.g., ML-KEM/ML-DSA) and required classical crypto functions; Validate RNG basics and	Enforce strong key lifecycle controls (HSM/TPM/cloud as applicable), M-of-N controls, downgrade	Validate crypto-agility and enterprise key management integration; confirm PQC conformanc	Mission-critical enhancements such as QKD/hybrid keying (if required), indigenous algorithms (not in GR scope), and	

		component inventory (BOM/CBOM).	resistance, and stronger entropy.	e and TRNG/QRNG integration	resilience/algorithm diversification.	
16	Interoperability	• Interoperability with Standardized APIs or reference implementation • Conformance with published RFCs by IETF of TCP/IP protocols (IPSec, TLS, HTTPS, API) • Validation of Hybrid implementations (Classical + PQC based implementations) • Cross-Library/Cross platform (Linux, windows etc.)/Cross language (C, Java etc.) Testing				
17	Performance Considerations	Establish baseline performance metrics (latency/throughput, keygen, encaps/decaps, sign/verify, hashing/HMAC).		Perform rigorous performance profiling (CPU/memory/power, scalability, bandwidth overhead) including agility/hybrid overhead.	Validate performance under stress/attack conditions plus disaster resilience/BCP requirements for critical services.	
	Security Assurance	Robustness against malformed inputs/forgery attempts; basic static vulnerability analysis.	Fuzzing/negative testing, VA/PT, secure coding evidence, Side-channel considerations; **For hardware:**	CI/CD security automation, Continuous vuln discovery, Audits, Supply-chain	Zero Trust failure-mode testing, Red teaming, Semi-Formal verification of critical components, Rigorous	

18			Secure boot/attestation/tamper resistance testing as applicable.	security, Secure KDF and centralized key management integration.	supply-chain assurance (down to semiconductor where needed), and Nation-state level threat simulation.	

4.3. Inspection to validate **Secure element, Trusted Execution Environment (TEE), Physically Unclonable Functions (PUFs), Secure boot attestation, tamper proof** as under:

4.3.1. Secure Element (SE)

4.3.1.1 **Objective**: Validate that the SE securely stores and processes cryptographic keys, and is resistant to physical and logical attacks.

4.3.1.2 **Testing and Validation Steps:**

Category	Test Activity	Description / Tools
Functional Tests	API compliance	Validate Global Platform or vendor API compliance (APDU command sequences).
	Key management	Test key generation, import/export, deletion policies, and secure lifecycle transitions.
	Cryptographic operations	Verify crypto operations using standard test vectors (NIST CAVP).
Security Tests	Access control enforcement	Validate PIN, password, or mutual authentication protection.
	Fault injection resilience	Perform voltage/clock glitch and EM fault tests to ensure resistance.
	Side-channel analysis	Conduct DPA/SPA, timing analysis, EMA and TVLA tests to measure leakage during crypto operations.
Certification Alignment	Common Criteria (CC) EAL 5+/FIPS 140-3 or equivalent	Check against CC Protection Profiles (e.g., PP0084 for SE).

4.3.2. Trusted Execution Environment (TEE)

4.3.2.1 **Objective**: Verify isolation, integrity, and trust chain between REE (Rich Execution Environment) and TEE.

4.3.2.2 **Testing and Validation Steps:**

Category	Test Activity	Description / Tools
Functional Tests	TEE Client-TA communication	Validate TEE Client API and Internal Core API compliance.
	Trusted App behavior	Verify secure storage, session management, and cryptographic functions inside TA.
Security Tests	Memory isolation	Confirm TEE memory isolation from REE via MMU configuration testing.
	Secure world boot & root of trust	Validate secure boot chain from ROM to TEE OS
	Access control	Test privilege escalation and shared memory vulnerabilities.
Certification Alignment	GlobalPlatform TEE PP	Validate compliance with TEE Protection Profile

4.3.3. Physically Unclonable Functions (PUFs)

4.3.3.1 **Objective**: Assess reliability, uniqueness, and tamper-resistance of PUF-derived keys or identifiers.

4.3.3.2 **Testing and Validation Steps:**

Metric	Description	Validation Method
Uniqueness	Different chips produce distinct responses.	Inter-chip Hamming Distance
Reliability (Stability)	Same chip produces same response under environmental variations.	Measure intra-chip HD under varying voltage, temp, aging.

Entropy and Randomness	Evaluate unpredictability of response bits.	NIST SP 800-22 randomness tests.
Tamper Resistance	PUF response alters irreversibly upon tampering.	Perform invasive probing, EM interference, decapsulation tests.
Reproducibility	Check if error correction mechanisms restore stable key.	Repeated power cycles and statistical validation.

4.3.4. Secure Boot & Attestation

4.3.4.1 **Objective**: Ensure only authenticated and unmodified firmware is executed and that device attestation is verifiable.

4.3.4.2 **Testing and Validation Steps:**

Category	Test Activity	Description
Functional Tests	Boot chain integrity	Validate each stage's digital signature verification (ROM ; Bootloader; OS).
	Firmware rollback prevention	Attempt to flash older firmware and check rejection.
Security Tests	Root of trust validation	Verify hash/signature against a known hardware root key.
	Remote attestation	Simulate verifier–prover exchange; validate attestation certificate and nonce freshness.
Tampering Tests	Modify bootloader or firmware	Confirm system refuses to boot untrusted images.
Standard Alignment	NIST SP 800-193, PSA Certified or equivalent	Check alignment with firmware protection and recovery guidelines.

4.3.5. Tamper-proof & Tamper Detection Mechanisms

4.3.5.1 **Objective**: Verify protection against physical attacks and that detection mechanisms respond correctly.

4.3.5.2 **Testing and Validation Steps:**

Type	Test Description	Expected Behavior
Active Tamper Detection	Simulate voltage, clock, or temperature anomalies.	Device triggers tamper interrupt, erases secrets.
Passive Tamper Resistance	Try to access protected areas via probing, fault injection.	No secret leakage; hardware protection active.
Packaging & Enclosure Tests	Apply mechanical stress, thermal cycling, microprobing.	Security mesh or coating triggers alerts.
Certification Mapping	FIPS 140-3 Level or equivalent	Validate against tamper-evident and tamper-response.

4.4. The product shall enforce and validate multi-person (M-of-N) authorization controls for all critical cryptographic operations, including master key generation, activation, and destruction. Validation shall confirm that operations cannot be executed without the required quorum, that minimum M and N values are configurable based on assurance level, and that single-person compromise is technically prevented. All multi-person control events shall be securely logged, auditable, and resistant to bypass or circumvention. The M-of-N quorum shall be mandatorily enforced for any changes to the PQC Transition Policy, including the switching of operating modes from "Hybrid" to "PQC-Only" or "Classical-Only."

4.5. The product shall enforce protocol-level protections against PQC parameter downgrade attacks, ensuring that adversaries cannot force negotiation of weaker security parameter sets when stronger options are available. Validation shall demonstrate strict enforcement of minimum approved parameter sets, rejection of downgrade attempts, immutable policy configuration, and conformance testing across supported protocols to ensure downgrade resistance cannot be bypassed.

4.6. The OEM shall ensure that **Vulnerability Assessment and Penetration Testing (VA/PT)** of the system is carried out by a designated security lab by NCCS or Information Security

Auditing Organization empanelled with CERT-In (MeitY, Government of India). For the cyber security audits that are conducted, ensure the adherence to latest guidelines issued by CERT-In or issued by Sectoral CERTs. The VA/PT report shall be submitted and reviewed for following:

4.6.1. Ensure testing covered all application components — API endpoints, web UI, backend services, and data interfaces.

4.6.2. Verify that cloud integrations and HSM interfaces were included in testing.

4.6.3. Ensure both automated and manual testing were performed.

4.6.4. Review the VA/PT report for:

- Classification of vulnerabilities (Critical, High, Medium, Low)
- Risk rating and CVSS scoring
- Recommended mitigations and closure evidence
- Mitigation Verification
- Check that all Critical and High vulnerabilities have been remediated and re-tested.

4.6.5. Validate that residual risk is documented and approved.

4.7. Secure Coding practice shall be verified with steps as under:

4.7.1. Static Analysis using tools.

4.7.2. Manual code inspection to verify:

- Input validation and sanitization
- Proper authentication and session management
- Secure cryptographic implementations (e.g., no hardcoded keys)
- Error/exception handling without information leakage
- Cryptographic Abstraction Layer-Verification shall confirm that all cryptographic functions are called via a standardized Abstraction Layer (API). Direct hard-coding of specific algorithms within application logic shall be prohibited.

4.7.3. The limited lists such as OWASP Top 10, SANS Top 25 and similar, should not be considered as standards or references. Instead, discovery of all known vulnerabilities should be based on the comprehensive standards/frameworks.

4.7.4. Check for vulnerabilities in third-party libraries

4.7.5. Confirm use of version control with restricted access (multi factor authentication)

4.7.6. Ensure code commits and merges require peer review and approval.

4.7.7. Ensure adherence to latest design/secure coding guidelines issued by CERT-In and applicable Sectoral CERTs.

CHAPTER-5

Quality, Safety, EMI/EMC and General Requirements

5.1. Quality requirements of a cryptographic system

5.1.1. The manufacturer shall furnish the MTBF values. A minimum value of MTBF shall be 10,000 hours. The calculations shall be based on the guidelines specified in the standard.

5.1.2. The product/systems shall be manufactured by the international quality management system ISO 9001:2015, for which the manufacturer should be duly accredited. A quality plan describing the manufacturer's quality assurance system must be submitted.

5.1.3. The product/systems shall conform to the requirements for the environment specified in document QM 333 {Latest issue: March 2010}: " Standard for environmental testing of Telecommunication Equipment" The applicable tests shall be for environmental category B2, including vibration test.

Quality requirements of a cryptographic system

Sl. No	Name of the Sub parameter	Description of Parameters and its range	Reference Standard (s)	Remarks
1	Operating Temperature	0°C to +60°C ~~and defence and space requirements shall work in the range -100°C to 200°C~~	IEC/ISO	For defence and space requirements to be met as per user specs.
2	Humidity	10 to 90% RH	IEC/ISO	
3	Reliability	Availability / operational uptime expressed as a percentage (e.g., ≥ 99.0%, 99.9%, 99.99%)	--	To be defined by procurer

4	Basic environmental Test	Environmental category B2	QM-333	
5	MTBF	Metric		At least 10,000 hours
6	MTTR	Metric		To be defined by procurer
7	Manufactured process compliance	International quality management	ISO 9001:2015	

5.2. EMI/EMC Requirements of a cryptographic system

The equipment shall conform to the EMC requirements as per the following standards and limits indicated therein. An accredited test agency shall furnish a test certificate and test report.

a) Conducted and radiated emission:
Name of EMC Standard: "CISPR 32 (2015) - Limits and methods of measurement of radio disturbance characteristics of Information Technology Equipment".

Limits: -

i. To comply with Class B limits of CISPR 32

ii. For Radiated Emission tests, limits below 1 GHz shall be as per relevant limits for measuring the distance of 10m OR as per relevant limits for measuring the distance of 3m.

b) Immunity to Electrostatic discharge:
Name of EMC Standard: IEC 61000-4-2 {2008) "Testing and measurement techniques of Electrostatic discharge immunity test".
Limits: -

i. Contact discharge level 2 {± 4 kV} or higher voltage;
ii. Air discharge level 3 {± 8 kV} or higher voltage;

c) Immunity to radiated RF:
Name of EMC Standard: IEC 61000-4-3 (2010) "Testing and measurement techniques-Radiated RF Electromagnetic Field Immunity test".

Limits: -

For Telecom Equipment and Telecom Terminal Equipment with Voice interface (s)

i. Under Test level 2 {Test field strength of 3 V/m} for general purposes in the frequency range 80 MHz to 1000 MHz and
ii. Under test level 3 (10 V/m) for protection against digital radio telephones and other RF devices in the frequency ranges 800 MHz to 960 MHz and 1.4 GHz to 6.0 GHz.
iii. For Telecom Terminal Equipment without Voice interface (s)
iv. Under Test level 2 {Test field strength of 3 V/m} for general purposes in the frequency range 80 MHz to 1000 MHz and for protection against digital radio telephones and other RF devices in frequency ranges 800 MHz to 960 MHz and 1.4 GHz to 6.0 GHz.

d) Immunity to fast transients (burst):
Name of EMC Standard: IEC 61000- 4- 4 {2012) "Testing and measurement techniques of electrical fast transients/burst immunity test".
Limits: -

Test Level 2, i.e., a) 1 kV for AC/DC power lines; b) 0. 5 kV for signal/control/ data/telecom lines;

e) Immunity to surges:
Name of EMC Standard: IEC 61000-4-5 (2014) "Testing & Measurement techniques for Surge immunity test".
Limits: -

i. For mains power input ports: (a)2 kV peak open circuit voltage for line-to-ground coupling (b) 1 kV peak open circuit voltage for a line-to-line coupling
ii. For telecom ports: (a) 2 kV peak open circuit voltage for a line to ground
iii. (b)2 kV peak open circuit voltage for a line-to-line coupling.

f) Immunity to conducted disturbance induced by Radiofrequency fields:
Name of EMC Standard: IEC 61000-4-6 (2013) "Testing & measurement techniques-Immunity to conducted disturbances induced by radio- frequency fields".

Limits: -

i. Under the test level 2 {3 V r.m.s.} in the frequency range 150 kHz-80 MHz for AC / DC lines and Signal /Control/telecom lines.

g) Immunity to voltage dips & short interruptions (applicable to AC/DC mains power input ports, if any):

Name of EMC Standard: IEC 61000-4-11/ IEC 61000-4-29 (2020) "Testing & measurement techniques- voltage dips, short interruptions and voltage variations immunity tests".

Limits: -

i. A voltage dip corresponding to a reduction of the supply voltage of 30% for 500ms (i.e., 70 % supply voltage for 500ms)
ii. A voltage dip corresponding to a reduction of the supply voltage of 60% for 200ms; (i.e., 40% supply voltage for 200ms)
iii. A voltage interruption corresponds to a reduction of a supply voltage of > 95% for 5s.
iv. A voltage interruption corresponds to a reduction of a supply voltage of >95% for 10ms.

Note 1: Classification of the equipment:

Class B: Class B is a category of apparatus that satisfies the class B disturbance Limits. Class B is intended primarily for use in the domestic environment and may include the following:

- Equipment with no fixed place of use; for example, portable equipment powered by built-in batteries;
- Telecommunication terminal equipment powered by the telecommunication networks
- Personal computers and auxiliary connected equipment

Please note that the domestic environment is an environment where the use of broadcast radio and television receivers may be expected within a distance of 10 m of the apparatus connected.

Class A: Class A is a category of all other equipment that satisfies the class A limits but not the class B limits.

Note 2: The testing agency for EMC tests shall be an accredited agency and details of accreditation shall be submitted.

Note 3: For checking compliance with the above EMC requirements, the method of measurements shall follow TEC Standard No. TEC/SD/DD/EMC-221/05/OCT-16 and the references mentioned therein unless otherwise specified. Alternatively, corresponding relevant Euro Norms of the above IEC/CISPR standards are also acceptable subject to the condition that frequency range and test level are met as per the above mentioned sub clauses (a) to (g) and TEC Standard No. TEC/SD/DD/EMC-221/05/OCT-16.

EMI/EMC requirements of a cryptographic system

Sl. No	Name of the Sub parameter	Types of Parameters range	Reference Standard(s)	Remarks
1	Conducted and radiated emission:		IEC CISPR 32 (2015) AMD1:2019	AC or DC supply voltage not exceeding 600 V
2	Immunity to Electrostatic discharge		IEC 61000-4-2 {2008)	static electricity discharges from operators directly and from personnel to adjacent objects
3	Immunity to radiated RF		IEC 61000-4-3 (2020)	
4	Immunity to fast transients (burst):		IEC 61000-4- 4 {2012)	
5	Immunity to surges:		IEC 61000-4-5 (2014)	

6	Immunity to conducted disturbance induced by Radio frequency fields:		IEC 61000-4-6 (2013)	Radiofrequency (RF) transmitters in the frequency range of 150 kHz up to 80 MHz
7	Immunity to voltage dips & short interruptions		IEC 61000-4-11 (2020) IEC 61000-4-29 (2020)	equipment with input current up to 16 A per phase

5.3. Safety Requirements of a cryptographic system

5.3.1. Electrical safety

IEC 62368-1 [replaced IS 13252-1/IEC 60950-1] is a primary reference for the safety of telecommunications equipment. Active electronics must comply with locally applicable electrical safety requirements in all cases. These safety parameters may include electrical insulation, grounding, fuses, current loss switches, etc. In case remote line powering is applied, it should comply with [ITU-T K.50], [ITU-T K.51] and [IEC 60950-21]. The safe working practices described in [ITU-T K.64] should be followed when work is carried out outside plant electronic equipment.

5.3.2. Laser safety

Since the box house active optical devices, it should comply with IEC 60825- 1 and IS 14624-2/IEC 60825-2 for optical safety requirements.

Note: This test shall be applicable if laser components are directly mounted in the box.

Table 8: Safety requirements of a cryptographic system

Sl. No	Name of the parameter	Description of Parameters and its range, if any	Reference Standard(s)	Remarks

1	Hazard-based product-safety standards for ICT and AV equipment	Audio/video, information and communication technology equipment - Part 1	IEC 62368-1: 2018 and COR1: 2020	Electrical safety for Hardware or S/W and or F/W over H/W
2	Safe limits for operating voltages and currents	telecommunication systems powered over the network	ITU-T K.50	Electrical safety for Hardware
3	safety criteria for telecommunicati on network equipment	requirements intended to reduce risks of fire, electric shock or injury	ITU-T K.51	persons who may come into contact with the equipment
4	Safe working practices for outside equipment installed In particular environments	working practices for service personnel to help them work safely in telecommunication installations	ITU-T K.64	The specific environments covered are characterized by wet conditions or close proximity to exposed metallic parts.
5	Information Technology Equipment – SAFETY	Remote power feeding	IEC 60950-21	Part 21 of IEC 60950
6	Safety of laser products emitting laser radiation	wavelength range 180 nm to 1 mm	IEC 60825- 1	Laser safety

7	Safetyof optical fibre communication systems (OFCSs)		IS 14624-2/IEC 60825-2	does not address safety issues associated with explosion or fire
8	Public safety: RoHS compliance	Safety from Hazardous material	EU 2015/863 Directive	restricts chemicals and heavy metals in electronic products

5.4. General Requirements

5.4.1. The system shall support In-field firmware upgrades from time to time for a continuation of functionality with the advancement of technology and interoperable and supporting systems to make it compatible.

5.4.2. It shall support remote system Software/Firmware upgrades.

5.4.3. As and when software bugs are found/ determined, the Manufacturer shall provide patches/firmware replacement, if involved, as mutually agreed between the Purchaser of the instrument and supplier. Modified documentation, wherever applicable, shall also be supplied.

5.4.4. The manufacturer/supplier shall furnish the list of recommended spares.

5.4.5. The supplier shall have a maintenance/repair facility in India. The supplier shall furnish MTBF and MTTR values.

5.4.6. The accessories cables shall have a low attenuation cable link, either optical or ethernet cable of the latest. The vendor will submit the Specification for the same.

5.4.7. It must be possible for an operator to select a particular encryption scheme for payload encryption system wide.

5.4.8. It shall automatically exchange a new session key on a pre-set interval of 1-60 minutes or user configured.

5.4.9. The new session key shall be generated automatically by a True Random Number Generator (TRNG) or a Pseudo Random Number Generator (PRNG). QRNGs are preferred over other TRNGs and PRNGs.

5.4.10. These devices should support high entropy throughput with very high randomness (entropy)

5.4.11. It shall provide confidentiality-protected firmware/software upgrades.

5.4.12. The encryption devices should be future-proof and fully reprogrammable for an upgrade to new algorithms based on the user requirements or availability of technology from time to time.

5.4.13. Cryptographic system can also support Quantum-safe key exchange algorithms under the standardisation process of NIST, along with classical algorithms in a hybrid manner.

5.4.14. Remote management should be possible only through secure Management software with minimum 2-factor authentication with hardware binding.

5.4.15. Cryptographic system shall support SNMPv3 or the latest and shall provide multiple manager support.

5.4.16. Cryptographic system shall support audit and event logging with Syslog support.

5.4.17. Cryptographic system shall be able to work with the NTP server for time synchronisation.

5.4.18. Cryptographic system shall be able to work with RADIUS or TACAS+ server for authentication

5.4.19. Repair procedure;

(i) List of replaceable parts used to include their sources and the approving authority.

(ii) Detailed ordering information for all the replaceable parts shall be listed in the manual to facilitate the reordering of spares as and when required.

(iii) A systematic procedure for troubleshooting and sub-assembly replacement shall be provided. Test fixtures and accessories required for repair shall also be indicated.

(iv) Systematic troubleshooting procedures shall be given for the probable faults with their remedial actions.

Note: The Purchaser may mention the repair manual requirement at the time of ordering.

5.4.20. Technical literature in Hindi or English of the instrument with block schematic

diagrams shall be provided. The complete layout and circuit diagrams of various assemblies with test voltages and waveforms at different test points of the units shall be provided, wherever required. All aspects of installation, operation, maintenance and repair shall be covered in the manuals. The soft copy/hard copy of the manuals shall also be provided. The manual shall include the following two parts:

(i) Installation, operation and maintenance manual.
(ii) Safety measures to be observed in handling the equipment.
(iii) Precautions for setting up, measurements and maintenance
(iv) Product/equipment required for routine maintenance and calibration, including their procedures.
(v) Illustration of internal and external mechanical parts.
(vi) A detailed description of the operation of the software used in the equipment, including its installation, loading and debugging etc.

5.4.21. Identification of Equipment

i) Equipment shall be marked with the supplier's or Manufacturer's logo/name.
ii) The Model No., serial No., The month and year of manufacture shall be indicated by screen printing on the body of the equipment or by a tamper-proof sticker pasted on the body of the equipment.
iii) Power Supply requirements shall be indicated on the body.
iv) The above markings shall be legible, indelible and easily visible

CHAPTER-6

Information for the procurer of the product

6.1. The procurer should require the vendor to submit Bill of Materials (BOM). The BOM submitted by the vendor should be in adherence to the SBOM, CBOM and HBOM guidelines issued by CERT-In or other sectoral CERTs.

6.2. The procurer should require the vendor to demonstrate cryptographic agility as an operational capability and not merely as design intent. The vendor shall provide documented procedures for algorithm addition, replacement, and deprecation without system downtime or architectural redesign.

6.3. The procurer should clearly specify acceptable cryptographic assurance mechanisms, such as compliance with ISO/IEC 19790, ISO/IEC 24759, FIPS 140-3, or Common Criteria.

6.4. The procurer should require that cryptographic policies (algorithm selection, key sizes, protocol versions) be configurable by the procuring entity and protected against unauthorized modification.

6.5. During migration phases, the procurer should mandate support for hybrid cryptographic modes (classical + PQC) with explicit controls to prevent downgrade attacks.

6.6. The procurer should require the vendor to disclose the performance, latency, power, and resource impact of post-quantum and hybrid cryptographic mechanisms under expected load conditions.

6.7. The procurer should define minimum support and maintenance periods for cryptographic components, including guaranteed availability of security updates until system end-of-life.

6.8. The below clauses from above chapters of GR are to be decided by procurer.

Sl. No.	GR clause / location (as written in draft)	What the procurer must decide
1.	Ch-4 Operational requirements table – "No of Concurrent connection" (Remarks: "Exact value to be finalized by procurer…")	Concurrency capacity sizing

2.	Ch-4 Operational requirements table – "Support of Jumbo frames" (Remarks: "exact maximum MTU to be specified by procurer")	Jumbo frame MTU requirement
3.	Ch-6 Quality requirements table – "Reliability (Availability/uptime)" (Remarks: "To be defined by procurer")	Minimum availability target
4.	Ch-6 Quality requirements table – "MTTR" (Remarks: "To be defined by procurer")	Maximum MTTR target
5.	General requirements – patch/firmware replacement ("…as mutually agreed between the Purchaser… and supplier")	Patch/SW update SLA and process
6.	General requirements – operator selection of encryption scheme ("must be possible… system wide")	Allowed algorithm/cipher-suite policy
7.	General requirements – session key change interval ("pre-set interval 1–60 minutes or user configured")	Rekey interval policy
8.	General requirements – RNG preference ("QRNGs are preferred…")	Required entropy source assurance
9.	Repair manual note ("Purchaser may mention the repair manual requirement at the time of ordering")	Whether repair manual is required
10.	Power table – AC supply connector type & redundancy (Remarks: "Procurer to specify connector type and redundancy…")	Power feed and redundancy
11.	Interoperability table – Clock accuracy ("accuracy requirements to be specified if needed")	Time/clock requirements

12.	Chapter-7 (existing text) – CBOM/BOM requirement, crypto agility, assurance mechanism, configurable policies, hybrid migration controls, performance disclosures, support period definition	Procurement-level compliance choices

DEFINITIONS AND TERMINOLOGY

Algorithm:

A specified mathematical process for computation; is a set of rules that, if followed, will give a prescribed result.

Application link:

A communication link is used to provide cryptographic applications in the user network.

Asymmetric key:

A cryptographic key is used with an asymmetric key (public key) algorithm. The Key may be a private key or a public key.

Authentication:

It is a property of an entity or party whose identity establish with a required assurance. The authenticated party could be a user, subscriber, home environment or serving network.

Approved:

Any authorised agency of Govt of India/FIPS approved and/or NIST-recommended.

Authentication protocol:

A defined sequence of messages between an entity and a verifier enables the verifier to perform authentication of an entity.

Authorisation:

The granting of rights, which includes granting access based on access rights.

Availability:

The property of an entity is accessible and useable upon demand by an authorised entity.

Credential:

A set of data presented as evidence of a claimed identity and/or entitlements.

Confidentiality:

The property that information is not made available or disclosed to unauthorised individuals, entities, or processes.

Communication channel:
Two communicating parties use that for exchanging data encoded in a form that may be non-destructively read and fully reproduced.

Certificate Revocation List (CRL):
A list of certificates revoked without expiry by a Certification Authority.

Certification Authority (CA):
The entity in a public key infrastructure (PKI) is responsible for issuing certificates to certificate subjects and exacting compliance with a PKI policy.

Ciphertext:
Data in its encrypted form.
Compromise:
The unauthorised disclosure, modification, substitution, or use of sensitive data (e.g., a secret key, private key, or secret metadata).

Confidentiality:
The property that sensitive information is not disclosed to unauthorised entities (i.e., the secrecy of key information is maintained).

Cross-certify:
Establishing a trust relationship between two Certification Authorities (CAs) by signing each other's public key in certificates is called a "cross-certificate."

Cryptographic algorithm:
A well-defined computational procedure that takes variable inputs, including a cryptographic key (if applicable), and produces an output.

Cryptographic boundary:
An explicitly defined continuous perimeter that establishes the physical bounds of a cryptographic module and contains all the hardware, software, and or firmware components of a cryptographic module.

Cryptographic checksum:
A mathematical value is created using a cryptographic algorithm assigned to data and later used to test the data to verify that the data has not changed.

Cryptographic hash function:
A function that maps a bit of arbitrary string length to a fixed-bit string length.

Approved hash functions satisfy the following properties:

1. One-way – Finding any input that maps to any pre-specified output is computationally infeasible.
2. Collision resistant – Finding two distinct inputs that map to the same output is computationally infeasible.

Cryptographic key:

A parameter used with a cryptographic algorithm determines its operation so that an entity with knowledge of the key can reproduce or reverse the process while an entity without knowledge of the key cannot. Examples include

1. The transformation of plaintext data into ciphertext data,
2. The transformation of ciphertext data into plaintext data,
3. The computation of a digital signature from data,
4. The verification of a digital signature,
5. The computation of a message authentication code (MAC) from data,
6. The verification of a MAC received with data,
7. The computation of a shared secret used to derive keying material.

Cryptographic primitive:

A low-level cryptographic algorithm is a fundamental building block for higher-level cryptographic algorithms. Cryptography is the discipline that embodies the principles, means, and methods for providing information security, including confidentiality, data integrity, source authentication, and non-repudiation.

Cryptoperiod:

When a specific key is authorised for use or in which the keys for a given system may remain in effect.

Data integrity:

A property whereby data has not been altered unauthorised since it was created, transmitted, or stored. Data integrity authentication: The process of determining the integrity of the data, also called integrity authentication or integrity verification.

Decryption:

The process of changing ciphertext into plaintext using a cryptographic algorithm and key.

Discrete Log Problem:

A mathematical problem is considered hard for a conventional computer to solve but is easily solved by a quantum computer. The problem requires an understanding of the concept of an algebraic group. Solve for k, where b^k=g and b and g are elements in the same algebraic group.

Digital signature:
The result of a cryptographic transformation of data that, when properly implemented, provides the services of NIST SP 800-175B

1. Source authentication,
2. Data integrity, and
3. Support for signer non-repudiation.

Digital Signature Algorithm (DSA):
A public key algorithm is used to generate and verify digital signatures.

Domain parameters:
The parameters used with a cryptographic algorithm are common to a domain of users.

Elliptic Curve Cryptography(ECC):
It is a type of public key cryptography; this acronym refers to a group of ciphers based on their security on the discrete logarithm problem over an elliptic curve cyclic group, i.e., a family of ciphers like ECDH, ECDSA and others.

Elliptic Curve Digital Signature Algorithm (ECDSA):
A digital signature algorithm that is an analogue of DSA using elliptic curves.

Encryption:
The process of changing plaintext into ciphertext using a cryptographic algorithm for security or privacy.

Entity:
An individual (person), organisation, device, or process. Ephemeral key pair A short-term key pair is used with a public key(asymmetric-key) algorithm that is generated when needed; the public key of a short key pair is not provided in a public key certificate, unlike static public keys, which are often included in a certificate.

Hash Function:

Used interchangeably with an algorithm in this document. Hash function See cryptographic hash function. Hash value results from applying a hash function to information, also called a message digest.

Identity authentication:
The process of assuring the identity of an entity interacting with a system; also see Source authentication.

Initialisation Vector (IV):
A vector is used in defining the starting point of a cryptographic process.

Integrity:
The property that Data has not been modified or deleted in an unauthorised and undetected manner.

Integrity authentication (integrity verification):
The process of determining the integrity of the data; is also called data integrity authentication.

Interoperability:
The ability of one entity to communicate with another entity. Key agreement A (pair-wise) key-establishment procedure where secret keying material is generated from information contributed by two participants so that no party can predetermine the value of the private keying material independently from the other party's contributions. Contrast with key-transport.

Key Confirmation:
A procedure assures one party that another possesses the same keying material and/or shared secret.

Key Derivation:
The process of keying material is derived from either a pre-shared key or a shared secret produced during a key-agreement scheme along with other information.

Key Establishment:
The procedure results in keying material that is shared among different entities.

Key Hierarchy:

A tree structure represents the relationship of different keys. In a key hierarchy, a node represents a key used to derive the keys the descendent nodes represent. A key can only have one precedent but may have multiple descendent nodes.

Keying material:

A cryptographic key and other parameters (e.g., IVs or domain parameters) are used with a cryptographic algorithm. When keying, the material is derived as specified in SP 800-56C4 and SP 800-108:5. Data is represented as a bit string such that any non-overlapping segments of the string with the required lengths can be used as secret keys, secret initialisation vectors, and other secret parameters.

Keying relationship, cryptographic:

The state exists between two entities, sharing at least one cryptographic Key.

Key Information:

Information related to a key includes the keying material and associated

metadata linking to that key.

Key Life Cycle:

A sequence of steps that a key undergoes from its reception by a key manager (KM) through its use in a cryptographic application and until deletion or preservation depending on the key management policy.

Key Management:

All activities performed on keys during their life cycle, starting from their reception from the quantum layer, storage, formatting, relay, synchronisation, authentication and supply to a cryptographic application and deletion or preservation, depending on the key management policy.

Key Manager (KM):

A functional module is located in a quantum key distribution (QKD) node to perform key management in the Key management layer.

Key Manager Link:

A communication link connecting key managers (KMs) to perform key management.

Key pair:
A public key and its corresponding private key; a key pair is used with a public key (asymmetric-key) algorithm

Key Relay: A method to share keys between arbitrary quantum key distribution (QKD) nodes via intermediate QKD node(s).

Key Symmetry: The key symmetry means that bit '0' and bit '1' probability detection should be nearly equal. NIST randomness test has to be performed on the raw key (bits detected by SPD) to validate the symmetry.

Key Supply: A function providing keys to cryptographic applications.

Key transport:
A key-establishment procedure whereby one party (the sender) selects a value for the secret keying material and then securely distributes that value to another party (the receiver). Contrast with a key agreement.

Key wrapping:
A method of cryptographically protecting the confidentiality and integrity of keys using a symmetric-key algorithm. Key-wrapping key A symmetric key provides confidentiality and integrity protection for other keys.

Merkle Tree:
A quantum-safe public key cryptography system based on a tree of message digests where each child leaf is computed using a cryptographic hash function that is keyed with a key derived from its parent.
Message Authentication Code (MAC):
A cryptographic checksum on data that uses an approved security function and a symmetric key to detect accidental and intentional modifications of data.

Message digest Metadata:
The information associated with a key describes its specific characteristics, constraints, acceptable uses, ownership, etc., sometimes called the key's attributes.

Mode of operation:
An algorithm that uses a block cipher algorithm as a cryptographic primitive to provide a cryptographic service, such as confidentiality or authentication.

Non-repudiation:
A service uses a digital signature that is used to support a determination of whether a given entity signed a message.

NP:

Class of computational decision problems for which any given yes-solution can be verified as a solution in polynomial time by a deterministic Turing machine (or solvable by a non-deterministic Turing machine in polynomial time).

NP-hard problem:
The problem X that we considered earlier should be as hard as every NP problem so that an easy solution for X will give an easy solution for every NP problem is called the NP-hard problem.

Network Function Virtualisation NFV:
Technology that enables the creation of logically isolated network partitions over shared physical networks so that heterogeneous collections of multiple virtual networks can simultaneously coexist over the shared networks.

One-time pad:
An unconditionally secure encryption method, where plaintext is encrypted with a random secret key(or pad) of the same length as the message. The Private Key must be known by the sender and receiver and used only once.
Owner of a certificate:
The entity that is responsible for managing the certificate, including requesting, replacing, and revoking the certificate if and when required. The certificate owner is not necessarily the subject entity associated with the public key in the certificate (i.e., the key pair owner).

Owner of a key or key pair:
One or more entities are authorised to use a symmetric key or the private key of a key pair.
Perfect Forward Secrecy:
An attribute of a security protocol that means that temporary/ephemeral cryptographic keys are used in the protocol so that if an adversary breaks the keys and can listen to traffic in the session, they can only listen for the current session and need to break the keys again in any future secure session.

Plaintext:

Data that has not been encrypted; intelligible data that has meaning and can be understood without decryption.

Pre-Shared Key:

A secret key that has previously been established between the parties who are authorised to use it by means of some secure method (e.g., using a secure manual distribution process or automated key-establishment scheme).

Polynomial Time:

A term used by computer scientists to describe the amount of computing time required to solve a mathematical problem as the problem scales upwards in size. A polynomial time algorithm means that the algorithm solves a problem very fast.

Privacy:

The right of individuals to control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.

Private key:

A cryptographic key is used with a public key cryptographic algorithm uniquely associated with an entity and not made public. In an asymmetric (public) key cryptosystem, the Private key is associated with a public key. Depending on the algorithm, the private key may be used to: -

i) Compute the corresponding public key,
ii) Compute a digital signature that the corresponding public key may verify.
iii) Decrypt data that was encrypted by the corresponding public key, or
iv) Compute a shared secret during a key-agreement process.

Protocol:

A set of rules used by two or more communicating entities that describe the message order and data structures for information exchanged between the entities.

Public key:

A cryptographic key is used with a public key (asymmetric key) algorithm uniquely associated with an entity that may be made public. In an asymmetric (public) key cryptosystem, the public key is associated with a private key. Anyone may know the public key and, depending on the algorithm may be used to -

1. Verify a digital signature signed by the corresponding private key.
2. Encrypt data that can be decrypted by the corresponding private key, or
3. Compute a shared secret during a key-agreement process.

Public key (Asymmetric-key) Cryptographic Algorithm:
A cryptographic algorithm that uses two related keys: a public key and a private key. The two keys have the property that determining the private key from the public key is computationally infeasible.

Public Key Infrastructure (PKI):
A framework is established to issue, maintain, and revoke public key certificates.
Quantum Channel: Communication channel for transmitting quantum signals.

Quantum-Safe Algorithm :

A step-by-step procedure that could run on a working quantum computer.

Quantum computing:

A computing device based on Qubits that can run the quantum computer.
Random Bit Generator (RBG):
A device or algorithm that outputs a sequence of bits that appears to be statistically independent and unbiased.

Relying party:
An entity that relies on the Certificate and the CA that issued the Certificate to verify the identity of the certificate owner, the validity of the public key, associated algorithms, and any relevant parameters in the Certificate, as well as the owner's possession of the corresponding private key.

RFC:

Request For Comment, which is a type of standard that the Internet Engineering Task Force publishes.
RSA:
A public key algorithm is used for key establishment and the generation and verification of digital signatures.

Scheme:
A set of unambiguously specified transformations that provide a (cryptographic) service (e.g., key establishment) when properly implemented and maintained. A scheme is a higher-level construct than a primitive and a lower-level construct than a protocol.
Secret key:

A single cryptographic key is used with a symmetric (secret key) cryptographic algorithm and is not made public (i.e., the key is kept secret). A private key is also called a symmetric key.

Sensitive (information):
Sensitive but unclassified information.
Security Association:
An instance of an encipherment key that temporarily protects network communications in an IPSec based VPN. An SA is a setup using the IKE protocol.
Security function: Cryptographic algorithms, together with modes of operation (if appropriate); for example, block cipher algorithms, digital signature algorithms, asymmetric key-establishment algorithms, message authentication codes, hash functions, or random bit generators.

Security strength:
A number is associated with the amount of work (i.e., the number of operations) required to break a cryptographic algorithm or system.

Sender/ Receiver:
This document defines the sender/transmitter and the receiver.
Shor's algorithm:
A method intended to run on a quantum computer that solves an instance of the Integer Factorization Problem and Discrete Log Problem in polynomial.
Signature Generation:
A digital signature algorithm and a private key generate a digital signature on data.
Signature Verification:
Using a digital signature and a public key to verify a digital signature on data.

Source Authentication:
The process of assuring the source of information is sometimes called data-origin authentication. Compare with Identity authentication.

SSL:

Secure Sockets Layer is an internet RFC that is a predecessor

Static Key Pair:

A long-term key pair for which the public key is often provided in a public key certificate.

Symmetric Key:

A single cryptographic key used with a symmetric (secret key) algorithm is uniquely associated with one or more entities and is not made public (i.e., the key is kept secret); a symmetric key is often called a secret key.

Symmetric-Key (Secret-Key) Algorithm:

A cryptographic algorithm that uses the same secret key for an operation and its complement (e.g., encryption and decryption).

TLS :

Transport Layer Security is an Internet RFC specifying a security protocol to encrypt and authenticate network communications for software applications. TLS v1.0 is the subsequent version of SSL v3.

Trusted Channel:

A channel where the endpoints are known and data integrity is protected in transit. Data privacy may be protected in transit depending on the communications protocol used. Examples include Transport Layer Security (TLS), IP security (IPSec), and secure physical connection.

User Network:

A network in which cryptographic applications consume keys supplied by a quantum key distribution (QKD) network or classical Key distribution network.

ACRONYMS

For this document the following abbreviations apply:

AC	Alternating Current
ACL	Access Control List
AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
AH	Authentication Header
ANS	American National Standard
ANSI	American National Standard Institute
CA	Certificate Authority
CBC	Cipher-Block Chaining
CFB	Cipher FeedBack mode
CLI	Command Line Interface
CMAC	Cipher-based Message Authentication Code
CNG	Cryptography API: Next Generation
CSR	Certificate Signing Requests
CTR	Counter
DC	Direct Current
DH	Diffie-Hellmen
DHKE	Diffie-Hellman Key Exchange
DSA	Digital Signature Algorithm
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EMI	electromagnetic Interference
EMC	Electromagnetic compatibility
ESP	Encapsulating Security Payload
FPGA	Field Programmable Gate Array
FTP	File Transfer Protocol
GCM	Galois/Counter Mode
HFE	Hidden Field Equations

HMAC	Hash-based Message Authentication Code
HTTPS	Hypertext Transfer Protocol Secure
IEC	International Electrotechnical Commission
IP	Internet Protocol
IPsec	IP security
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IKE	Internet Key Exchange
IKEv2	Internet Key Exchange version 2
ITU	International Telecommunication Union
IV	Initialisation Vector
KMAC	Keccak Message Authentication Code
KME	Key Management Entity
KMF	Key Management Framework
KMIE	Key Management Interoperability Protocol
LWE	Learning With Error
MAC	Message Authentication Code
NIST	National Institute of Standards and Technology
OASIS	Organization for the Advancement of Structured Information Standards
OFB	Output FeedBack mode
OID	Object Identifier
OSI	Open Systems Interconnection
PFS	Perfect Forward Secrecy
PKI	Public Key Infrastructure
PQC	Post-Quantum Cryptography
PRNG	Pseudo Random Number Generator
QKD	Quantum Key Distribution
QKDE	Quantum Key Distribution Entity
RADIUS	Remote Authentication Dial-In User Service
REST	REpresentational State Transfer
RH	Relative Humidity
RFC	Request For Comment

RSA	Rivest, Shamir and Adleman
SA	Security Associations
SAE	Secure Application Entity
SIS	Short Integer Solution
SFP	Small Form-factor Pluggable
S/MIME	Secure/Multipurpose Internet Mail Extention
SNMP	Simple Network Management Protocol
SP	Special Publication
SSH	Secure Shell
SSL	Secure Sockets Layer
SVP	Shortest Vector Problem
TLS	Transport Layer Security
TRNG	True Random Number Generator
TACAS	Terminal Access Controller Access Control System
USB	Universal Serial Bus
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
WAN	Wide Area Network
XMSS	eXtended Merkle Signature Scheme

======End of the document ======